**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Môn học: Phát triển ứng dụng web**

**ĐỀ TÀI**

**XÂY DỰNG TRANG WEB BÁN HÀNG COOL MATE**

**Giảng viên hướng dẫn: Ths. Phạm Thị Kim Ngoan**

**Sinh viên thực hiện: Trần Duy Phát**

**Mã số sinh viên: 63135054**

**Lớp: 63. CNTT-4**

**Mục lục**

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 4](#_Toc152495240)

[1.1. Giới thiệu đề tài: 4](#_Toc152495241)

[1.2. Mục tiêu của đề tài: 4](#_Toc152495242)

[1.3. Giới hạn và phạm vi của đề tài: 5](#_Toc152495243)

[1.4. Các yêu cầu đối với website: 5](#_Toc152495244)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 7](#_Toc152495245)

[2.1. Tổng quan về ngôn ngữ Asp.NET MVC 7](#_Toc152495246)

[2.1.1. Giới thiệu Asp.NET MVC Framework 7](#_Toc152495247)

[2.1.2 Lý do chọn Asp.NET MVC 12](#_Toc152495248)

[2.2. Tổng quan về SQL SERVER 13](#_Toc152495249)

[2.2.1. Hệ quản trị CSDL SQL Server 13](#_Toc152495250)

[2.2.2. Đặc điểm của SQL Server và đối tượng làm việc 14](#_Toc152495251)

[CHƯƠNG 3: WEBSITE/ KẾT QUẢ PHÁT TRIỂN CỦA WEBSITE 15](#_Toc152495252)

[3.1 Thiết kế cơ sở dữ liệu 15](#_Toc152495253)

[3.1.1 Thiết kế cơ sở dữ liệu của website 15](#_Toc152495254)

[3.1.2 Mô hình cơ sở dữ liệu 16](#_Toc152495255)

[3.1.3 Danh sách các lớp đối tượng 16](#_Toc152495256)

[3.5 Giao diện phần mền 23](#_Toc152495257)

[3.5.1 Giao diện trang Admin 23](#_Toc152495258)

[3.5.2 Giao diện trang người dùng 30](#_Toc152495259)

[CHƯƠNG 4: KẾT LUẬN 35](#_Toc152495260)

[4.1 Kết quả đạt được của đề tài 35](#_Toc152495261)

[4.2 Hạn chế của đề tài 35](#_Toc152495262)

[4.3 Kết luận 35](#_Toc152495263)

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu đề tài:

Cool Mate là một cửa hàng thời trang chuyên bán cho lứa tuổi từ 20 trở lên. Nơi này có những sản phẩm như là đồ thể thao, đồ văn phòng, đồng phục và cả phụ kiện. Với phong cách đa dạng. Với phong cách thiết kế đa dạng, từ trang phục công sở đến trang phục dạo phố, từ phong cách thanh lịch đến phong cách trẻ trung, cửa hàng đáp ứng nhu cầu của các khách hàng với sự đa dạng và phong phú của sản phẩm.

Cửa hàng Cool Mate luôn chú trọng đến chất lượng sản phẩm và sự thoải mái của khách hàng khi dùng đến đồ của cửa hàng. Ngoài ra, cửa hàng còn cam kết bán với giá xứng đáng với những gì khách hàng mua được và còn được đổi trả miễn phí.

Trước đây Cool Mate chỉ tập trung bán cho các đối tượng xung quanh mình, nhưng với sự phát triển của mạng truyền thông như Facebook, Instagram hay là các shop online,… Với nhu cầu mở rộng thị trường đẩy mạnh sản phẩm và dễ dàng cho việc quản lý cửa hàng vì thế việc lựa chọn có một website riêng mang tính đặc trưng cho cửa hàng là một nhu cầu tất yếu.

## Mục tiêu của đề tài:

Phạm vi nghiên cứu của đề tài tập trung vào các nội dung sau đây:

* Tìm hiểu nhu cầu của các cửa hàng thời trang đối với việc xây dựng website quản lý và bán hàng trực tuyến.
* Đánh giá các yếu tố ảnh hưởng đến trải nghiệm người dùng và hiệu quả kinh doanh của website, bao gồm: giao diện, tốc độ tải trang, tính năng, bảo mật và tiếp thị trực tuyến.
* Thiết kế và phát triển website quản lý cửa hàng thời trang với các tính năng quản lý đơn hàng, lưu trữ thông tin khách hàng và quản lý sản phẩm và dịch vụ.
* Áp dụng các chiến lược tiếp thị trực tuyến để quảng bá sản phẩm và thu hút khách hàng, đánh giá hiệu quả của các chiến lược này.
* Nghiên cứu và áp dụng các kỹ thuật mới nhất để tối ưu hóa website, giảm thời gian tải trang và tăng trải nghiệm người dùng.
* Đo lường và đánh giá hiệu quả của website, đảm bảo sự hoạt động ổn định và liên tục cập nhật để đáp ứng nhu cầu của khách hàng.

## Giới hạn và phạm vi của đề tài:

* Phạm vi nghiên cứu của đề tài tập trung vào các nội dung sau đây:
* Tìm hiểu nhu cầu của các cửa hàng thời trang đối với việc xây dựng website quản lý và bán hàng trực tuyến.
* Đánh giá các yếu tố ảnh hưởng đến trải nghiệm người dùng và hiệu quả kinh doanh của website, bao gồm: giao diện, tốc độ tải trang, tính năng, bảo mật và tiếp thị trực tuyến.
* Thiết kế và phát triển website quản lý cửa hàng thời trang với các tính năng quản lý đơn hàng, lưu trữ thông tin khách hàng và quản lý sản phẩm và dịch vụ.
* Áp dụng các chiến lược tiếp thị trực tuyến để quảng bá sản phẩm và thu hút khách hàng, đánh giá hiệu quả của các chiến lược này.
* Nghiên cứu và áp dụng các kỹ thuật mới nhất để tối ưu hóa website, giảm thời gian tải trang và tăng trải nghiệm người dùng.
* Đo lường và đánh giá hiệu quả của website, đảm bảo sự hoạt động ổn định và liên tục cập nhật để đáp ứng nhu cầu của khách hàng.

## Các yêu cầu đối với website:

Một trang web bán hàng hiệu quả cần đáp ứng rất nhiều yêu cầu để tạo ra trải nghiệm tích cực cho người dùng và đảm bảo mang lại lợi ích cho doanh nghiệp. Dưới đây là một số yêu cầu:

* Giao diện người dùng:
  + Phải thân thiện với người dùng, trực quan.
  + Thiết kế giao diện phải tương thích được với nhiều thíết bị.
  + Tốc độ tải trang nhanh
* Quản lý sản phẩm:
  + Thiết kế trực quan, dễ quản lý
  + Hiển thị hình ảnh chất lượng cao, mô tả chi tiết sản phẩm
  + Danh mục sản phẩm rõ ràng
  + Hỗ trợ quản lý hàng tồn kho
* Quy trình thanh toán và đặt hàng:
  + Hệ thống thanh toán an toàn và đa dạng (thẻ tín dụng, chuyển khoản, ví điện tử).
  + Quy trình đặt hàng đơn giản và dễ theo dõi.
* Bảo mật:
  + Giao tiếp an toàn qua HTTPS.
  + Bảo vệ dữ liệu cá nhân của khách hàng.
  + Tích hợp các biện pháp bảo mật để ngăn chặn gian lận.
* Tính năng tìm kiếm:
  + Hệ thống tìm kiếm hiệu quả với khả năng lọc kết quả.
  + Tính năng gợi ý sản phẩm.
* Tương tác và phản hồi:
  + Hệ thống đánh giá và nhận xét sản phẩm.
  + Hỗ trợ trực tuyến hoặc chatbot để giải đáp thắc mắc của khách hàng.
* Chăm sóc khách hàng:
  + Hỗ trợ trực tuyến, qua điện thoại hoặc email.
  + Hệ thống theo dõi và giải quyết khiếu nại.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về ngôn ngữ Asp.NET MVC

Mô hình kiến trúc Model-View-Controller (MVC) được áp dụng để chia ứng dụng thành ba thành phần chính: model, view và controller. ASP.NET MVC là một nền tảng giúp tạo ra các ứng dụng web sử dụng mô hình MVC thay vì theo mô hình ASP.NET Web Forms. ASP.NET MVC được đánh giá vì tính nhẹ, dễ kiểm thử giao diện (so với Web Forms), và khả năng tích hợp tốt với các tính năng của AưSP.NET. Nó được định nghĩa 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 quen thuộc. Một số ứng dụng web phù hợp với kiến trúc MVC, trong khi những ứng dụng khác có thể ưa chuộng ASP.NET Web Forms và cơ chế postbacks. Đôi khi, có các ứng dụng kết hợp cả hai kiến trúc để đáp ứng yêu cầu cụ thể của họ.

2.1.1. Giới thiệu Asp.NET MVC Framework

ASP.NET là một nền tảng ứng dụng web (web application framework) được phát triển và cung cấp bởi Microsoft, cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web. - Dựa trên ASP.NET, ASP.NET MVC cho phép các nhà phát triển phần mềm xây dựng một ứng dụng web dựa trên mẫu thiết kế MVC. - MVC là một mẫu thiết kế (design pattern) chuẩn, được sử dụng nhằm chia ứng dụng thành ba thành phần chính: model, view và controller. - Phiên bản ASP.NET MVC: ASP.NET MVC 5 (10/2013), ASP.NET MVC 5.2.7 (11/2017).

Nền tảng MVC bao gồm các thành phần dưới đây :

*Hình 1: Mẫu Model - View - Controller*

**Models:** Các đối tượng Models là một phần của ứng dụng, thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu, và cập nhật dữ liệu trở lại vào bảng Products ở SQL Server.

Trong các ứng dụng nhỏ, model thường chỉ là một khái niệm nhằm phân biệt, chứ không nhất thiết phải được cài đặt thực tế. Ví dụ, nếu ứng dụng chỉ đọc dữ liệu từ CSDL và gửi chúng đến view, không cần phải có tầng model và các lớp liên quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model).

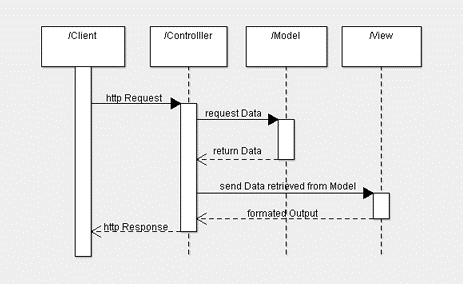
**Views:**  là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Product.

**Controllers**: Controller là các thành phần dùng để quản lý tương tác người dùng, làm việc với model và chọn view để hiển thị giao diện người dùng. Trong một ứng dụng MVC, view chỉ được dùng để hiển thị thông tin, controller chịu trách nhiệm quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng. Ví dụ, controller sẽ quản lý các dữ liệu người dùng gởi lên (query-string values) và gởi các giá trị đó đến model, model sẽ lấy dữ liệu từ CSDL nhờ vào các giá trị này.

Mẫu MVC giúp bạn tạo được các ứng dụng mà chúng phân tách rạch ròi các khía cạnh của ứng dụng (logic về nhập liệu, logic xử lý tác vụ và logic về giao diện). Mẫu MVC chỉ ra mỗi loại logic kể trên nên được thiếp lập ở đâu trên ứng dụng. Logic giao diện (UI logic) thuộc về views. Logic nhập liệu (input logic) thuộc về controller. Và logic tác vụ (Business logic – là logic xử lý thông tin, mục đích chính của ứng dụng) thuộc về model. Sự phân chia này giúp bạn giảm bớt được sự phức tạp của ứng dụng và chỉ tập trung vào mỗi khía cạnh cần được cài đặt ở mỗi thời điểm. Ví dụ như bạn chỉ cần tập trung vào giao diện (views) mà không phải quan tâm đến logic xử lý thông tin của ứng dụng.

Để quản lý sự phức tạp của ứng dụng, mẫu MVC giúp cho chúng ta có thể kiểm thử ứng dụng dễ dàng hơn hẳn so với khi áp dụng mẫu Web Forms. Ví dụ, trong một ứng dụng ASP.NET Web Forms, một lớp thường được sử dụng để hiển thị thông tin xuất ra cho người dùng và đồng thời xử lý thông tin người dùng nhập. Việc xây dựng các bộ test tự động cho ứng dụng Web Forms là rất phức tạp, bởi để kiểm thử mỗi trang web, bạn phải khởi tạo đối tượng trang, khởi tạo tất cả các control được sử dụng trong trang và các lớp phụ thuộc trong ứng dụng. Và bởi vì có quá nhiều lớp cần được khởi tạo để chạy được trang, thật khó để có thể viết các test chỉ tập trung vào một khía cạnh nào đó của ứng dụng. Và vì thế, kiểm thử đối với các ứng dụng dứa trên nền tảng Web Forms sẽ khó khăn hơn nhiều so với khi áp dụng trên ứng dụng MVC. Hơn thế nữa, việc kiểm thử trên nền tảng Web Forms yêu cầu phải sử dụng đến web server. Nền tảng MVC phân tách các thành phần và sử dụng các interface (khái niệm giao diện trong lập trình hướng đối tượng), và nhờ đó có thể kiểm thử các thành phần riêng biệt trong tình trạng phân lập với các yếu tố còn lại của ứng dụng

Sự phân tách rạch ròi ba thành phần của ứng dụng MVC còn giúp cho việc lập trình diễn ra song song. Ví dụ như một lập trình viên làm việc với view, lập trình viên thứ hai lo cài đặt logic của controller và lập trình viên thứ ba có thể tập trung vào logic tác vụ của model tại cùng một thời điểm.



Hình 2: Mô hình tuần tự của MVC

Lấy ví dụ một GUI Component (thành phần đồ họa người dùng ) đơn giản là checkbox. Checkbox có thành phần Model để quản lý trạng thái của nó là check hay uncheck, thành phần View để thể hiện nó với trạng thái tương ưng lên màn hình, và thành phần Controller để xử lý nhưng sự kiện khi có sự tương tác của người sử dụng hoặc các đối tượng khác lên checkbox.

Khi ngươi sử dụng nhần chuột vào Check box , thành phần Controller của Checkbox sẽ xử lý sự kiện này, yêu cầu thành phần Model thay đổi dữ liệu trạng thái. Sau khi thay đổi trạng thái, thành phần Model phát thông điệp đến thành phần View và Controller. Thành phần View của Checkbox nhận được thông điệp sẽ cập nhật lại thể hiện của Checkbox, phản ánh chính sác trạng thái Checkbox do Model lưu giữ. Thành phần Controller nhận được thông điệp do Model gởi tới sẽ có nhưng tương tác phản hòi với người sử dụng nếu cần thiết.

**Tiềm hiểu thêm về Controler**

Controller có trách nhiệm chính là điều hướng các yêu cầu của người sử dụng. Như vậy trên toàn ứng dụng của ta, tất cả các request đều sẽ phải đi tới controller. Và tại đây, ứng với các tham số người sử dụng truyền mà ta đưa họ đến một tác vụ nào đó trên ứng dụng.

Tại các tác vụ này, chúng sẽ thông qua lớp model để làm việc và trả kết quả trở về controller. Cuối cùng controller sẽ đẩy dữ liệu thao tác tới view. View là thành phần cuối cùng mà người sử dụng nhận được khi họ giở request tới ứng dụng.

Có thể hiểu controller, giống với kỹ thuật đa cấp với các tác vụ chạy ứng dụng phân cấp theo từng nhánh riêng biệt như: Module, action,… Qua hình vẽ này, ta hiểu rằng. Để có thể thao tác với các action (hành động) ta cần phải đi qua file index.php. Lúc này file index đóng vai trò như một controller được dùng để điều hướng các request. Ứng với các request thì nó sẽ trả về một controller khác để xử lý tác vụ một cách cụ thể. Tại controller con, nó sẽ gọi các action riêng biệt. Ở đó, thông qua các action mà nó gọi tới các file xử lý giản đơn.

**Tìm hiểu Model**

Model là thành phần chủ yếu được sử dụng để thao tác xử lý dữ liệu. Trong các framework, Model vẫn thường sử dụng theo phương thức Active Record. Một trong những design pattern. Chúng có tác dụng rút ngắn thời gian viết câu truy vấn cho người sử dụng. Biến những câu truy vấn phức tạp trở nên gần gũi và thân thiện với người sử dụng thông qua các thư viện được định nghĩa sẵn.

Model thường sẽ là các phương thức có trách nhiệm xử lý các tác vụ như: select, insert, update, delete các record trong database. Ứng với các lấy dữ liệu, model thường sử dụng mảng để gởi trả kết quả về. Vì mảng có thể cho phép model lưu trữ nhiều thông tin hơn, nên thường các record khi bóc tách chúng sẽ mang các dữ liệu của database một cách chi tiết.  
Khi sử dụng models, ta cũng cần tuân theo nguyên tắc chính của chúng là không xuất giá trị trực tiếp trong model. Mà tất cả những dữ liệu ấy, phải đưa vào mảng và trả về theo phương thức. Và tiếp tục ở view ta sẽ sử dụng nó để lấy dữ liệu ra.

**Tìm hiều về View**

View là phần hiển thị thông tin tương phản khi gởi và nhận request. Trước đây, khi người lập trình chưa nghĩ tới view. Họ thường thao tác xử lý dữ liệu ngay trực tiếp trên ứng dụng và đổ cả dữ liệu ngay trên file PHP đó. Điều này làm cho ứng dụng trở nên cồng kềnh, và đặc biệt rất khó cho việc bảo trì nâng cấp sau này. Nhất là đối với designer, việc thay đổi giao diện của một website luôn làm cho họ cảm thấy đau đầu vì phải vọc thẳng vào core. Trước đây, để giải quyết tình huống này. Người ta thường sử dụng template để phân tách website thành 2 mảng riêng biệt. Một là giao diện và một là core. Việc chỉnh sửa giao diện trở nên đơn giản hơn đối với họ so với cách viết thập cẩm kia. Tuy nhiên, các thư viện này thực chất sẽ làm cho ứng dụng của chúng ta trở nên chậm chạp hơn bao giờ hết. Bởi chúng phải phiên dịch nhiều lần các kịch bản.

Chẳng hạn: Để dễ thao tác, smarty sẽ dịch ngược các yêu cầu của bạn sang ngôn ngữ của nó. Sau đó chúng sẽ chuyển ngôn ngữ đó sang PHP và thao tác xử lý trên nó.  
Việc này sẽ làm ứng dụng chậm chạp, do cứ phải dịch qua, dịch lại một kịch bản. Trong khi, với sự kết hợp của PHP thuần, ứng dụng của bạn sẽ nhanh và ổn định hơn nhiều. Và view cũng là một phần trong việc nâng cấp những hạn chế ấy. Chúng giúp giảm thiểu tối đa quá trình biên dịch nhiều lần. Và làm cho ứng dụng trở nên mạnh mẽ và chuyên nghiệp hơn nhiều so với cách lập trình thuần.

2.1.2 Lý do chọn Asp.NET MVC

* Các tính năng của Asp.NET MVC:
* 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 (.mater) Bạn có thể dụng các tình năng có sẵn của ASP.NET như dùng 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, seession và profile, quản lý tình trạng ứng dụng, hệ thống cấu hình…
* ASP.NET MVC5 còn 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.

2.2. Tổng quan về SQL SERVER

2.2.1. Hệ quản trị CSDL SQL Server

Khái niệm hệ quản trị cơ sở dữ liệu SQL Server

SQL Server là một hệ thống quản lý cơ sở dữ liệu RDBMS sử dụng Transact-SQL để trao đổi dữ liệu giữa Client computer và SQL Server computer. Một RDBMS bao gồm Databases, Database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera -Byte và có thể phục vụ cùng lúc cho hàng ngàn User. SQL Server 2000 có thể kết hợp "ăn ý" với các Server khác như Microsoft Internet Information Server (IIS), ECommerce Server, Proxy Server...

**2.2.2. Đặc điểm của SQL Server và đối tượng làm việc**

* SQL là ngôn ngữ tựa tiếng Anh.
* SQL là ngôn ngữ phi thủ tục, nó không yêu cầu ta cách thức truy nhập CSDL như thế nào. Tất cả các thông báo của SQL đều rất dễ sử dụng và ít khả năng mắc lỗi .
* SQL cung cấp tập lệnh phong phú cho các công việc hỏi đáp dữ liệu
* Chèn, cập nhật, xoá các hàng trong một quan hệ.
* Tạo, sửa đổi, thêm và xoá các đối tượng trong của CSDL.
* Điều khiển việc truy nhập tới cơ sở dữ liệu và các đối tượng của CSDL để đảm bảo tính bảo mật của cơ sở dữ liệu.
* Đảm bảo tính nhất quán và sự ràng buộc của CSDL.
* Yêu cầu duy nhất để sử dụng cho các hỏi đáp là phải nắm vững được các cấu trúc CSDL của mình.

# CHƯƠNG 3: WEBSITE/ KẾT QUẢ PHÁT TRIỂN CỦA WEBSITE

3.1 Thiết kế cơ sở dữ liệu

3.1.1 Thiết kế cơ sở dữ liệu của website

Với một hệ thống bán hàng để đầy đủ về mặt cơ sở dữ liệu cần các bảng như sau:

* Categories (Id, Name, Slug, ParentId, Order, MetaDesc, MetaKey, Create By, Create At, Update By, Update At, Status)
* Suppliers (Id, Name, Image, Slug, Order, FullName, Phone, Email, UrlSite, MetaDesc, MetaKey, Create At, Update By, Update At, Status)
* Products (Id, CatID, Name, Supplier, Slug, Image, Price, SalePrice, Amount, MetaDesc, MetaKey, Create At, Update By, Update At, Status)
* Menus (Id, Name, TableId, TypeMenu, Position, Link, ParentId, Order, Create At, Update By, Update At, Status)
* Posts (Id, TopId, Title, Slug, Detail, Image, PostType, MetaDesc, MetaKey, Create At, Update By, Update At, Status)
* Links (Id, Slug, TableId, Type)
* Topics (Id, Name, Slug, ParentId, Order, MetaDesc, MetaKey, Create At, Update By, Update At, Status)
* Sliders (Id, Name, URL, Image, Order, Position, MetaDesc, MetaKey, Create At, Update By, Update At, Status)
* Orders (Id, UserID, ReceiverAddress, Note, Create At, Update By, Update At, Status)
* OrderDetails (Id, OrderID, ProductId, Price, Amount)
* Contacts (Id, UserID, FullName, Phone, Email, Title, Detail, Create At, Update By, Update At, Status)

3.1.2 Mô hình cơ sở dữ liệu

A screenshot of a computer

Description automatically generated

###### ***Hình 4: Biểu đồ dữ liệu quan hệ***

3.1.3 Danh sách các lớp đối tượng

1. ***Lớp Categories (Lớp loại sản phẩm)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | **Id** | Int | Khóa chính | Mã loại sản phẩm |
| **2** | Name | Nvarchar(max) |  | Tên loại sản phẩm |
| **3** | Slug | Nvarchar(max) |  | Tên rút gọn |
| **4** | ParentId | Int |  | Cấp cha |
| **5** | Order | Int |  | Sắp xếp |
| **6** | MetaDesc | Nvarchar(max) |  | Mô tả |
| **7** | MetaKey | Nvarchar(max) |  | Từ khóa |
| **8** | CreateBy | Int |  | Người tạo |
| **9** | CreateAt | Datetime |  | Ngày tạo |
| **10** | UpdateBy | Int |  | Người cập nhật |
| **11** | UpdateAt | Datetime |  | Ngày cập nhật |
| **12** | Status | Int |  | Trạng thái |

*Bảng 3-1: Danh sách các thuộc tính lớp categories*

1. ***Lớp Suppliers (Lớp nhà cung cấp)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | **Id** | Int | Khóa chính | Mã nhà cung cấp |
| **2** | Name | Nvarchar(max) |  | Tên nhà cung cấp |
| **3** | Image | Nvarchar(max) |  | Hình ảnh |
| **4** | Slug | Nvarchar(max) |  | Tên rút gọn |
| **5** | Order | Int |  | Sắp xếp |
| **6** | FullName | Nvarchar(max) |  | Tên đầy đủ |
| **7** | Phone | Nvarchar(max) |  | Số điện thoại |
| **8** | Email | Nvarchar(max) |  | Thư điện tử |
| **9** | UrlSite | Nvarchar(max) |  | Link liên kết |
| **10** | MetaDesc | Nvarchar(max) |  | Mô tả |
| **11** | MetaKey | Nvarchar(max) |  | Từ khóa |
| **12** | CreateBy | Int |  | Người tạo |
| **13** | CreateAt | Datetime |  | Ngày tạo |
| **14** | UpdateBy | Int |  | Người cập nhật |
| **17** | UpdateAt | Datetime |  | Ngày cập nhật |
| **18** | Status | Int |  | Trạng thái |

*Bảng 3-2: Danh sách các thuộc tính lớp Suppliers*

1. ***Lớp Products (Lớp sản phẩm)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | **Id** | Int | Khóa chính | Mã sản phẩm |
| **2** | CatID | Int |  | Mã loại sản phẩm |
| **3** | Name | Nvarchar(max) |  | Tên sản phẩm |
| **4** | Supplier | Int |  | Mã nhà cung cấp |
| **5** | Slug | Nvarchar(max) |  | Tên rút gọn |
| **6** | Image | Nvarchar(max) |  | Hình ảnh |
| **7** | Price | Decimal (18,2) |  | Giá bán |
| **8** | SalePrice | Decimal (18,2) |  | Giá giảm |
| **9** | Amount | Int |  | Số lượng |
| **10** | MetaDesc | Nvarchar(max) |  | Mô tả |
| **11** | MetaKey | Nvarchar(max) |  | Từ khóa |
| **12** | CreateBy | Int |  | Người tạo |
| **13** | CreateAt | Datetime |  | Ngày tạo |
| **14** | UpdateBy | Int |  | Người cập nhật |
| **15** | UpdateAt | Datetime |  | Ngày cập nhật |
| **16** | Status | Int |  | Trạng thái |

*Bảng 3-3: Danh sách các thuộc tính lớp Products*

1. ***Lớp Menus***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | **Id** | Int | Khóa chính | Mã menu |
| **2** | Name | Int |  | Tên menu |
| **3** | TableId | Nvarchar(max) |  | Bảng dữ liệu |
| **4** | TypeMenu | Nvarchar(max) |  | Kiểu Menu |
| **5** | Position | Nvarchar(max) |  | Vị trí |
| **6** | Link | Nvarchar(max) |  | Địa chỉ |
| **7** | ParentId | Nvarchar(max) |  | Cấp cha |
| **8** | Order | Nvarchar(max) |  | Sắp xếp |
| **9** | CreateBy | Nvarchar(max) |  | Người tạo |
| **10** | CreateAt | Int |  | Ngày tạo |
| **11** | UpdateBy | Datetime |  | Người cập nhật |
| **12** | UpdateAt | Int |  | Người cập nhật |
| **13** | Status | Datetime |  | Trạng thái |

*Bảng 3-4: Danh sách các thuộc tính lớp Menus*

1. ***Lớp Posts (Lớp bài viết)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | **Id** | Int | Khóa chính | Mã bài viết |
| **2** | TopID | Int |  | Chủ đề bài viết |
| **2** | Title | Nvarchar(max) |  | Tên bài viết |
| **3** | Slug | Nvarchar(max) |  | Tên rút gọn |
| **4** | Detail | Nvarchar(max) |  | Chi tiết bài viết |
| **5** | Image | Nvarchar(max) |  | Hình ảnh |
| **6** | PostType | Nvarchar(max) |  | Kiểu bài viết |
| **7** | MetaDesc | Nvarchar(max) |  | Mô tả |
| **8** | MetaKey | Nvarchar(max) |  | Từ khóa |
| **9** | CreateBy | Int |  | Người tạo |
| **10** | CreateAt | Datetime |  | Ngày tạo |
| **11** | UpdateBy | Int |  | Người cập nhật |
| **12** | UpdateAt | Datetime |  | Ngày cập nhật |
| **13** | Status | Int |  | Trạng thái |

*Bảng 3-5: Danh sách các thuộc tính lớp Posts*

1. ***Lớp Links (Lớp liên kết)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | **Id** | Int | Khóa chính | Mã bài viết |
| **2** | Slug | Nvarchar(max) |  | Tên rút gọn |
| **3** | TableId | Int |  | Tên bài viết |
| **4** | Type | Nvarchar(max) |  | Loại liên kết |

*Bảng 3-6: Danh sách các thuộc tính lớp Links*

1. ***Lớp Topics (Lớp chủ đề)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | **Id** | Int | Khóa chính | Mã chủ đề |
| **2** | Name | Nvarchar(max) |  | Tên chủ đề |
| **3** | Slug | Nvarchar(max) |  | Liên kết |
| **4** | ParentId | Int |  | Cấp cha |
| **5** | Order | Int |  | Sắp xếp |
| **6** | MetaDesc | Nvarchar(max) |  | Mô tả |
| **7** | MetaKey | Nvarchar(max) |  | Từ khóa |
| **8** | CreateBy | Int |  | Người tạo |
| **9** | CreateAt | Datetime |  | Ngày tạo |
| **10** | UpdateBy | Int |  | Người cập nhật |
| **11** | UpdateAt | Datetime |  | Ngày cập nhật |
| **12** | Status | Int |  | Trạng thái |

*Bảng 3-7: Danh sách các thuộc tính lớp Topics*

1. ***Lớp Sliders***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | **Id** | Int | Khóa chính | Mã slider |
| **2** | Name | Nvarchar(max) |  | Tên slider |
| **3** | URL | Nvarchar(max) |  | Liên kết |
| **4** | Image | Nvarchar(max) |  | Hình ảnh |
| **5** | Order | Int |  | Sắp xếp |
| **6** | Position | Nvarchar(max) |  | Vị trí |
| **7** | MetaDesc | Nvarchar(max) |  | Mô tả |
| **8** | MetaKey | Nvarchar(max) |  | Từ khóa |
| **9** | CreateBy | Int |  | Người tạo |
| **10** | CreateAt | Datetime |  | Ngày tạo |
| **11** | UpdateBy | Int |  | Người cập nhật |
| **12** | UpdateAt | Datetime |  | Ngày cập nhật |
| **13** | Status | Int |  | Trạng thái |

*Bảng 3-8: Danh sách các thuộc tính lớp Sliders*

1. ***Lớp Orders (Lớp đơn đặt hàng)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | **Id** | Int | Khóa chính | Mã đơn đặt hàng |
| 2 | UserID | Nvarchar(max) |  | Mã khách hàng |
| 3 | ReceiverAddress | Nvarchar(max) |  | Địa chỉ nhận hàng |
| 4 | Note | Nvarchar(max) |  | Ghi chú |
| 5 | CreateBy | Int |  | Người tạo |
| 6 | CreateAt | Datetime |  | Ngày tạo |
| 7 | UpdateBy | Int |  | Người cập nhật |
| 8 | UpdateAt | Datetime |  | Ngày cập nhật |
| 9 | Status | Int |  | Trạng thái |

*Bảng 3-9: Danh sách các thuộc tính lớp Orders*

1. ***Lớp OrderDetails (Lớp chi tiết đơn hàng)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | **Id** | Int | Khóa chính | Mã đơn hàng |
| **2** | OrderID | Int |  | Tên chủ đề |
| **3** | ProductId | Int |  | Liên kết |
| **4** | Price | Decimal (18,2) |  | Cấp cha |
| **5** | Amount | Decimal (18,2) |  | Sắp xếp |

*Bảng 3-10: Danh sách các thuộc tính lớp OrderDetails*

1. ***Lớp Contact (Lớp liên hệ)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | **Id** | Int | Khóa chính | Mã chủ đề |
| **2** | UserID | Int |  | Mã khách hàng |
| **3** | FullName | Nvarchar(max) |  | Tên đầy đủ |
| **4** | Phone | Nvarchar(max) |  | Số điện thoại |
| **5** | Email | Nvarchar(max) |  | Thư điện tử |
| **6** | Title | Nvarchar(max) |  | Tiêu đề |
| **7** | Detail | Nvarchar(max) |  | Chi tiết |
| **8** | CreateBy | Int |  | Người tạo |
| **9** | CreateAt | Datetime |  | Ngày tạo |
| **10** | UpdateBy | Int |  | Người cập nhật |
| **11** | UpdateAt | Datetime |  | Ngày cập nhật |
| **12** | Status | Int |  | Trạng thái |

*Bảng 3-11: Danh sách các thuộc tính lớp Contact*

3.5 Giao diện phần mền

3.5.1 Giao diện trang Admin

Trang giao diện khi người quản lý mới đăng nhập ở trang đăng nhập dành cho admin vào bằng tài khoản admin.

A screenshot of a computer

Description automatically generated

**Hình 3.2. Trang Index của Admin**

A screenshot of a computer

Description automatically generated

**Hình 3.3. Giao diện khi nhấn vào ô “sản phẩm”, dropdown được thả xuống**

A screenshot of a computer

Description automatically generated

**Hình 3.4. Giao diện khi ấn vào “Bài viết”, dropdown được thả xuống.**

A screenshot of a menu

Description automatically generated

**Hình 3.5. Giao diện khi ấn vào “Giao diện”, dropdown được thả xuống.**

A screenshot of a computer

Description automatically generated

**Hình 3.6. Giao diện ở trang Index của Product**

Trang index dùng để hiển thị tất cả sản phẩm được người quản lý tạo ra từ trang Create.cshtml (Tạo mới sản phẩm). Dữ liệu được hiển thị lấy từ model ở bảng Products với status là 1 hoặc 2 (1 là trạng thái hiển thị và 2 là trạng thái không hiển thị trong View ở trang chủ (Site)). Với 2 nút lần lượt ở góc phải trên cùng màn hình là “Thêm” và “Thùng rác” thì nút “Thêm” là chuyển sang trang Create.cshtml khi muốn thêm sản phẩm và nút “Thùng rác” để vào trang Trash.cshtml khi muốn xem các sản phẩm ở thùng rác. Ở đây có các nút ở ô “Chức năng” với các chức năng lần lượt từ trái sang phải là chuyển trạng thái, cập nhật sản phẩm, xem chi tiết sản phẩm và cuối cùng là xóa sản phẩm ra khỏi trang Index. Với chức năng chuyển trạng thái thì trạng thái của sản phẩm được chuyển qua lại lần lượt là 1 hoặc 2 (nghĩa là nếu ở 2 thì thành 1 và ngược lại). Chức năng cập nhật sản phẩm là chuyển vào trang Edit.cshtml của Product để chỉnh sửa thông tin của sản phẩm. Còn với chức năng xem chi tiết, thì khi nhấn vào sẽ chuyển sang trang Detail.cshtml của Product để xem thông tin chi tiết của sản phẩm đấy. Cuối cùng là chức năng xóa sản phẩm, thì đây là chuyển trạng thái từ 1 hay 2 thành 0 để chuyển sản phẩm đó sang trang Trash.cshtml để thực hiện xóa sản phẩm.

A screenshot of a computer

Description automatically generated

**Hình 3.7. Giao diện ở trang create của product**

Các dữ liệu như loại sản phẩm là được truy vấn từ bảng Categories trong model để hiển thị ra các loại sản phẩm đang có ở trong cơ sở dữ liệu hay như tên nhà cung cấp là được lấy từ bảng Suppliers trong model để hiển thị ra các nhà cung cấp hiện có trong cơ sở dữ liệu, nút “Quay lại” ở góc bên phải trên cùng có nhiệm vụ là để quay lại trang Index.cshtml của Product và nút “Chọn tệp” để tải ảnh sản phẩm lên. Dữ liệu từ View Create.cshtml do admin nhập thông tin vào sẽ được gửi yêu cầu xử lý cho Controller thông qua HTTPPost. Sau khi kiểm tra dữ liệu được nhập vào phù hợp với model thì chuyển sang cho model lưu dữ liệu. Còn trường hợp dữ liệu sai thì Controller gửi thông báo lỗi cho View để hiển thị lỗi.

A screenshot of a computer

Description automatically generated

**Hình 3.8. Giao diện ở trang Trash của Product**

Trang trash hiển thị các sản phẩm trong cơ sở dữ liệu từ model các sản phẩm có trạng thái (status) là 0. Với ô chức năng từ trái sang phải gồm có nút “undo” và “trash”. Click vào nút có hình “undo” thì sản phẩm sẽ được trả lại trạng thái là 1 và được hiển thị lại trong view ở trang Index của Product. Còn khi nhấn vào nút “trash” thì View gửi yêu cầu xử lý cho controller thông qua HttpPost để controller chuyển tới trang Delete.cshtml để thực hiện việc xóa sản phẩm ra khỏi cơ sở dữ liệu.

A screenshot of a computer

Description automatically generated

**Hình 3.9. Giao diện ở trang Delete của Product**

Giao diện ở trang Delete để xác thực rằng người quản lý có chắc chắn xóa sản phẩm ra khỏi cơ sở dữ liệu hay không. Nếu không thì nhấn vào nút “Quay lại” để trở về trang Trash.cshtml. Còn nếu chắc chắn thì nhấn vào nút “Xóa [Delete]” để tiến hành xóa sản phẩm ra khỏi cơ sở dữ liệu. Khi nút được ấn thì View gửi yêu cầu xử lý cho controller để controller xóa sản phẩm và ảnh của sản phẩm được lưu trong thư mục ra khỏi cơ sở dữ liệu và file ảnh được lưu trong thư mục ảnh của product.

A screenshot of a computer

Description automatically generated

**Hình 3.10. Giao diện ở trang Edit của Product**

Giao diện của trang Edit sẽ thực hiện như trang Create nhưng sẽ khác ở chỗ là trang Edit.cshtml cần phải xóa ảnh của sản phẩm đã có trước đây ra khỏi cơ sở dữ liệu và thư mục ảnh để tiến hành lưu ảnh mới của sản phẩm vào.

**A screenshot of a website

Description automatically generatedHình 3.11. Giao diện ở trang Detail của Product**

Giao diện để hiển thị chi tiết toàn bộ sản phẩm, với các nút lần lượt ở trên là “Sửa” và “Quay lại”. Chức năng “Sửa” dùng để chuyển sang trang Edit.cshtml đối với trường hợp người quản lý muốn sửa thông tin của sản phẩm và chức năng “Quay lại” nếu muốn ra khỏi trang Detail để chuyển về lại Index

Giao diện của trang Product sẽ tương tự như các trang Supplier (Danh mục nhà cung cấp), trang Category (Danh mục loại sản phẩm), trang Post (Danh sách bài viết), trang Topic (Chủ đề bài viết), trang Page (Trang đơn).

3.5.2 Giao diện trang người dùng

Trang giao diện khi người dung truy cập vào trang web bán hàng

A person in a black shirt

Description automatically generated

Thanh main menu có drop list hiện ra các loại sản phẩm chi tiết khi hover vào các loại sản phẩm hiện trên thanh main menu

A screenshot of a computer

Description automatically generated

Phần slider có thể chuyển ảnh hình tự động hoặc người dùng có thể tự chuyển ảnh

A person in a white shirt

Description automatically generated

Phần hiện các sản phẩm được các sản phẩm có bán được admin mở lên

A screenshot of a clothing store

Description automatically generated

Giao diện khi người dùng bấm vào loại sản phẩm

A screenshot of a website

Description automatically generated

A screenshot of a clothing store

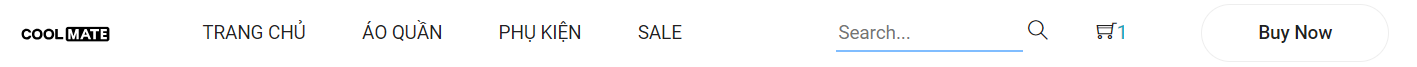
Description automatically generated

Giao diện khi người bấm vào các sản phẩm để hiện ra chi tiết của sản phẩm người dùng muốn xem

A person wearing a green jacket

Description automatically generated

Khi bấm vào thêm vào giỏ hàng sản phẩm sẽ được thêm vào giỏ hàng thì tại icon giỏ hàng sẽ tang từ 0 lên bằng số sản phẩm bạn thêm vào giỏ.



Thanh tìm kiếm để tìm kiếm sản phẩm các sản phẩm có trong shop

A screenshot of a computer

Description automatically generated

Nếu tìm thấy sản phẩm giao diện sẽ hiện ra các sản phẩm có từ khóa mà người tìm kiếm tại thanh tìm kiếm

A screenshot of a website

Description automatically generated

Và giao diện khi người dùng bấm và giỏ hàng, tại đây người dùng có thể thêm sản phẩm, khi người dùng mua sản phẩm số lượng nhiều sẽ bấm cập nhât lại đơn hàng, người dùng có thể xóa đơn hàng và cuối cùng là thanh toán.

A screenshot of a website

Description automatically generated

# CHƯƠNG 4: KẾT LUẬN

4.1 Kết quả đạt được của đề tài

* Xây dựng website bán hàng
* Giao diện đẹp, màu sắc hài hòa, dễ nhìn, dễ sử dụng.
* Xây dựng được các chức năng cần thiết cho hệ thống: thêm, sửa, xóa thông tin.

4.2 Hạn chế của đề tài

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

4.3 Kết luận

Trong quá trình nguyên cứu và vận dụng các kiến thức đã được học. Chúng em đã hoàn thành phần mềm ”bán hàng” nhằm mục đích hoạt động kinh doanh và ôn lại những kiến thức đã học. Chúng em hy vọng sau hoạt động nhóm này là tiền đề để chúng em phát triển mạnh hơn về kiến thức và chất lượng phần mềm.

Trong quá trình hoàn thành phần mềm với kinh nghiệm thực tế chưa nhiều nên báo cáo không tránh khỏi những sai sót, rất mong sự góp ý của thầy.

Chúng em xin chân thành cảm ơn.

**TÀI LIỆU THAM KHẢO**

* Phạm Thị Kim Ngoan, Slides Bài giảng Phát triển ứng dụng Web,2023.
* Bùi Chí Thành , Giáo trình Phát triển ứng dụng web.
* Đoàn Vũ Thịnh, Bài hướng dẫn thực hành Phát triển ứng dụng web.

**Source code:** [**https://github.com/DuyPhatpeo/PTUDW**](https://github.com/DuyPhatpeo/PTUDW)