BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

---🙢🕮🙠---



**BÀI TẬP LỚN**

**XÂY DỰNG TRANG WEB BÁN HÀNG AROMA**

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

**Sinh viên thực hiện: Nguyễn Thiết Duy Đan**

**Mã số sinh viên: 63133615**

**Lớp: 63. CNTT-4**

**Khánh Hoà – tháng 12/2023**

Mục lục

[Chương 1. TỔNG QUAN VỀ ĐỀ TÀI 2](#_Toc152427878)

[1.1 giới thiệu về shop aroma 2](#_Toc152427879)

[1.2 mục tiêu 2](#_Toc152427880)

[1.3 phạm vi 3](#_Toc152427881)

[1.4 các yêu cầu đối với website 3](#_Toc152427882)

[Chương 2. Cơ sở lý thuyết 4](#_Toc152427883)

[2.1 tổng quan về lập trình web 4](#_Toc152427884)

[2.2 mô hình asp.net mvc framework 4](#_Toc152427885)

[2.2.1 Giới thiệu ASP.NET MVC 4](#_Toc152427886)

[2.3 cơ sở dữ liệu sql server 9](#_Toc152427887)

[2.3.1 Hệ quản trị cơ sở dữ liệu SQL Server 9](#_Toc152427888)

[2.3.2 Đặc điểm của SQL Server 9](#_Toc152427889)

[Chương 3. Kết quả xây dựng website “Aroma Shop” 10](#_Toc152427890)

[Chương 4. Kết luận 10](#_Toc152427891)

[TÀI LIỆU THAM KHẢO 11](#_Toc152427892)

1. TỔNG QUAN VỀ ĐỀ TÀI
   1. giới thiệu về shop aroma

Aroma là một cửa hàng thời trang chuyên bán cho lứa tuổi từ 20 trở lên. Nơi này có những sản phẩm như là giày, dép, 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. Ngoài ra, Aroma Shop cũng cung cấp những mặt hàng khác như đồng hồ, khăn và chăn mềm mang đến cho khách hàng sự tiện lợi khi có thể tìm thấy tất cả các mặt hàng cần thiết tại một địa điểm duy nhất.

Cửa hàng Aroma 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 Aroma 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.

* 1. mục tiêu

Áp dụng các kiến thức đã được học để xây dựng phát triển một website chuyên nghiệp và hiệu quả đầy đủ chức năng, giúp quản lý bán hàng trực tuyến cho cửa hàng, cụ thể như sau:

* Thiết kế và phát triển một giao diện website đẹp và thân thiện với người dùng, giúp khách hàng dễ dàng tìm kiếm và mua sản phẩm trực tuyến.
* Tối ưu hóa website để đảm bảo tốc độ tải trang nhanh, giảm thời gian chờ đợi và tăng trải nghiệm người dùng.
* Đảm bảo việc bảo mật thông tin khách hàng, bảo vệ dữ liệu và giảm thiểu rủi ro về an ninh mạng.
* Cung cấp các tính năng quản lý đơn hàng, lưu trữ thông tin khách hàng và quản lý sản phẩm và dịch vụ.
* 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.
  1. phạm vi

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

* 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.
  1. 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.
* Tính năng tìm kiếm:
  + Hệ thống tìm kiếm hiệu quả với khả năng lọc kết quả.
  + Tính năng gợi ý sản phẩm.

1. Cơ sở lý thuyết
   1. tổng quan về lập trình web

**Lập trình web** là công việc có nhiệm vụ nhận tất cả các dữ liệu từ bộ phận **thiết kế web**để chuyển thành một bộ máy web hoàn chỉnh có tác động qua lại với cơ sở dữ liệu và tương tác với người dùng dựa trên ngôn ngữ máy tính. Một khi xây dựng trang web xong thì lập trình viên có thể được phân công để quản trị website, lúc đó lập trình viên cần trang bị thêm một số công cụ nhằm giúp cho việc quản trị trở nên đơn giản hơn. Các công cụ đó sẽ hỗ trợ cho bạn kiểm tra được những lần uptime, downtime, phần trăm thoát trang web, nguồn traffic đổ vào web, hoặc tình trạng quá tải băng thông, …Tuy nhiên, đó sẽ là công việc sau này, bạn có thể vừa học cách lập trình web cơ bản, đồng thời bổ sung thêm kiến thức quản trị website vào thời gian rảnh rỗi.

* 1. mô hình asp.net mvc framework
     1. Giới thiệu ASP.NET MVC

ASP.NET (ASP - Active Server Pages, NET - Network Enabled Technologies) được biên dịch dưới dạng **Common Language Runtime (CLR)**, là một nền tảng dành cho phát ứng dụng web động. Các ứng dụng ASP.Net được viết bằng nhiều loại ngôn ngữ .Net khác nhau, phổ biến như C #, VB.Net và J #. ASP.NET gồm 2 model là Web Forms và MVC.

ASP.NET MVC là một [framework](https://vietnix.vn/framework-la-gi/) được phát triển bởi Microsoft để xây dựng ứng dụng web dựa trên [mô hình MVC](https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/) (Model-View-Controller). Đây là một phần của nền tảng ASP.NET, được thiết kế để tạo ra các ứng dụng web linh hoạt, dễ bảo trì và kiểm thử. Phiên bản ASP.NET MVC: ASP.NET MVC 5 (10/2013), ASP.NET MVC 5.2.7 (11/2017).

MVC là viết tắt của “Model-View-Controller,” một mô hình thiết kế phần mềm được sử dụng để phân chia logic và trách nhiệm trong ứng dụng. Mô hình này giúp tổ chức mã nguồn một cách có tổ chức và dễ quản lý.

Mẫu Model – Controller – View

**Model**: Đại diện cho dữ liệu và logic xử lý dữ liệu của ứng dụng. Model đóng vai trò như một kho chứa dữ liệu để có thể truy cập và lấy dữ liệu ra.

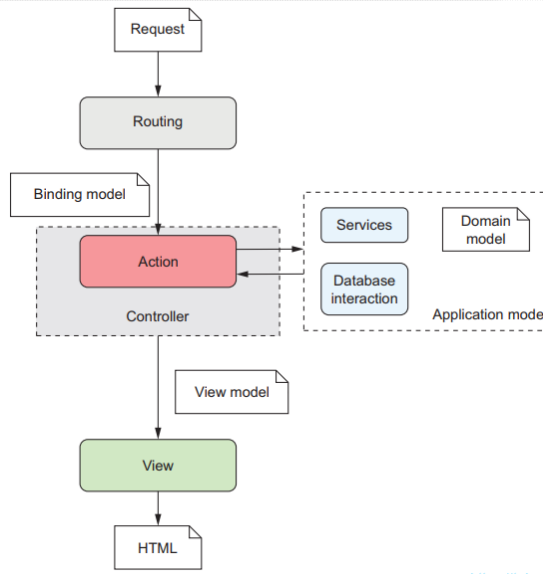
**View**: Là thành phần hiển thị lên trang web giao diện người dùng (UI), nó dựa vào các thông tin dữ liệu từ model để đưa lên web.

**Controller**: Là thành phần chứa logic điều khiển. Nó xử lý các sự kiện và tương tác của người dùng, sau đó gọi các phương thức tương ứng trong Model để thay đổi dữ liệu và cập nhật View. Controller làm nhiệm vụ điều phối giữa Model và View.

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.

Mô hình tuần tự của MVC

Truy vấn HTTP được cơ chế Routing ánh xạ sang phương thức gọi là action. Các class action được chứa trong Controller, action chứa các code C# xử lí dữ liệu và những dữ liệu đó được đưa đến HTML thông qua View. View là các trang Razor chưa các mã C# + Html (Razor View).

Ví dụ như là lấy một giao diện người dùng cơ bản là checkbox thì Checkbox có thành phần Model để lưu trữ trạng thái là đã uncheck hay check với value là 1 và 0, thành phần View sẽ hiển thị kết quả của nó lên trên giao diện và thành phần Controller để xử lí sự kiện là người dùng có tác động lên checkbox.

Khi người dùng tác động lên checkbox thì controller sẽ xử lí sự kiện này, rồi yêu cầu thay đổi trạng thái của Model. Sau khi thay đổi trạng thái thì Model đã phát tín hiệu đến View và Controller. Thành phần View sau khi nhận được tín hiệu thay đổi đến từ Model thì sẽ cập nhật để thể hiện lên giao diện để người dùng nhìn thấy, còn thành phần Controller khi nhận được tín hiệu sẽ thực hiện các thay đổi cũng như cập nhật hay là phản hồi lại cho người dùng nếu như cần thiết.

**Tìm hiểu về Controller**

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 đây các tác vụ sẽ được xử lí thông qua các lớp Model. Các tác vụ sẽ trao đổi qua lại với lớp Model và trả trở về lại controller. Sau đó nó sẽ được controller đẩy dữ liệu đến View. View chính là thành phần cuối cùng mà người dùng có thể thấy được khi mà họ đưa ra request. Trong mỗi controller sẽ có các action chịu trách nhiệm xử lí các request đó. Mỗi controller sẽ có nhiều action và mỗi action là một chức năng tương ứng.

**Tìm hiểu về 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.

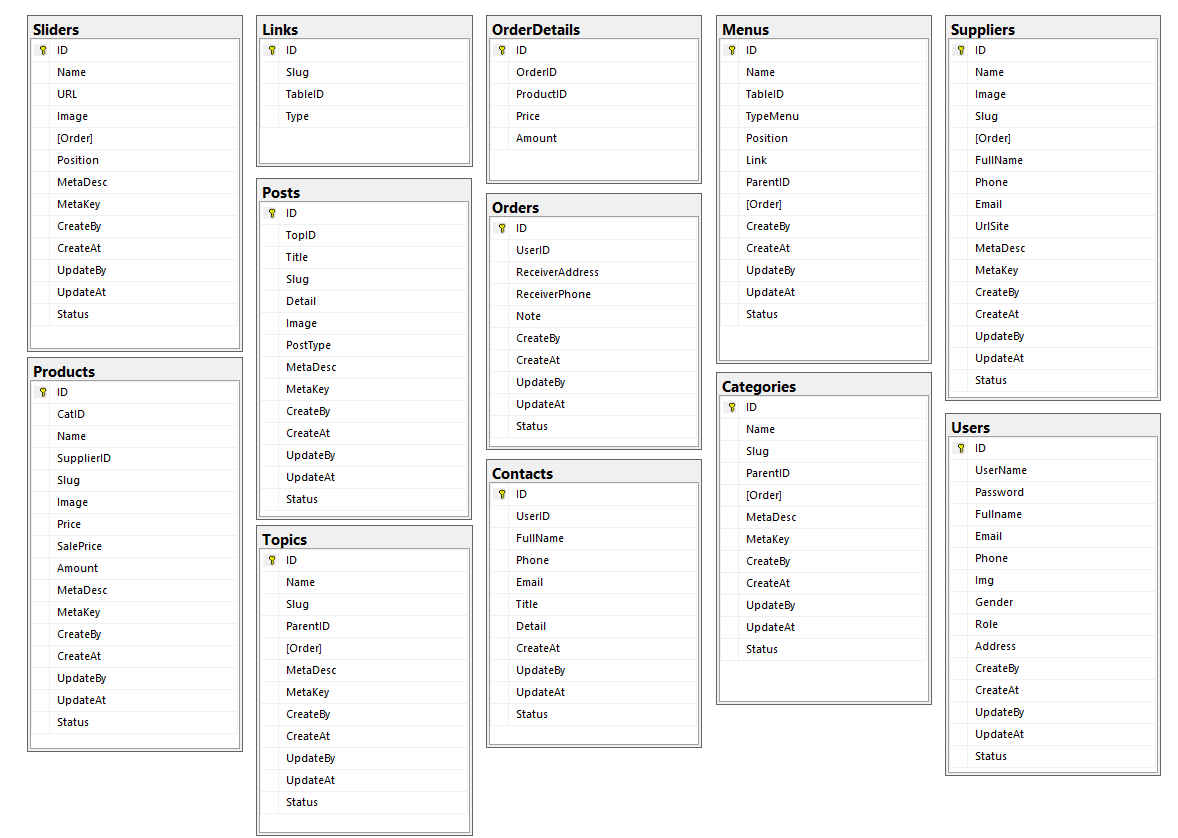
**Tìm hiểu về View**

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.

* 1. cơ sở dữ liệu sql server
     1. Hệ quản trị cơ sở dữ liệu SQL Server

Hệ quản trị cơ sở dữ liệu (Database Management System) có thể hiểu là hệ thống được thiết kế để quản lí một khối lượng dữ liệu nhất định một cách tự động và có trật tự. Các hành động quản lý này bao gồm chỉnh sửa, xóa, lưu thông tin và tìm kiếm (truy xuất thông tin) trong một nhóm dữ liệu nhất định.

* + 1. Đặc điểm của SQL Server
* HQTCSDL cung cấp giao diện lập trình dễ sử dụng và ngôn ngữ tương tác với dữ liệu được lưu trữ bên trong CSDL
* Sử dụng ngôn ngữ tiếng anh
* 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.

1. Kết quả xây dựng website “Aroma Shop”
   1. cơ sở dữ liệu của website

Các bảng của cơ sở dữ liệu

* + 1. Mô tả chi tiết từng bảng:
       1. Lớp Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | ID | int | Khóa chính | Mã để phân biệt |
| **2** | Name | nvarchar(MAX) |  | Tên loại sản phẩm |
| **3** | Slug | nvarchar(MAX) |  | Tên rút gọn |
| **4** | ParenltID | 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 |

* + - * 1. Các thuộc tính của Categories
      1. Lớp Suppliers
      2. Lớp Products
      3. Lớp Orders
      4. Lớp Users
      5. Lớp Posts
      6. Lớp Links
      7. Lớp Topics
      8. Lớp Menus
      9. Lớp OrderDetails
      10. Lớp Contact
  1. các đối tượng sử dụng:
* Đối với admin: Admin sử dụng trang admin, có quyền (thêm/xóa/sửa/phục hồi) các sản phẩm, loại sản phẩm, nhà cung cấp, các chủ đề, trang đơn và quản lí các tài khoản người dùng (Tài khoản Admin thì không chỉnh được còn tài khoản người dùng là (Xóa/Phục hồi)).
* Đối với user: User sử dụng trang site, có quyền xem sản phẩm, mua sản phẩm và đăng nhập/đăng kí.
  1. các chức năng chính của từng website:
  2. trang web dành cho admin
     1. Quản lí
  3. trang web dành cho user

1. Kết luận

TÀI LIỆU THAM KHẢO

* Phạm Thị Kim Ngoan, *Giáo trình Phát triển ứng dụng web*
* MVP Lê Hoàng Dũng (2/12/2023), *Tổng quan và giới thiệu về ASP.NET MVC,* [*https://support.microsoft.com/vi-vn/topic/31846479-c656-f2a4-bc24-c9803a97e62c*](https://support.microsoft.com/vi-vn/topic/31846479-c656-f2a4-bc24-c9803a97e62c)