**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**----🙣🕮🙡----**



**TIỂU LUẬN CHUYÊN NGÀNH**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE QUẢN LÝ SPA**

**GVHD : TS. Nguyễn Thiên Bảo**

**SVTH : Lê Minh Phước 15110103**

**Vũ Ngọc Phát 15110096**

**Tp. Hồ Chí Minh, Tháng 12 năm 2019**

 CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

Tp. Hồ Chí Minh, ngày tháng 9 năm 2019

|  |  |
| --- | --- |
| **NHIỆM VỤ TIỂU LUẬN CHUYÊN NGÀNH** |  |
| Họ và tên sinh viên: Lê Minh Phước MSSV: 15110103 | Lớp:15110CL1A |
| Vũ Ngọc Phát MSSV: 15110096 | Lớp: 15110CL2A |

Ngành: Công nghệ thông tin

Giảng viên hướng dẫn: TS. Nguyễn Thiên Bảo ĐT: 0927613761

Ngày nhận đề tài: 31/08/2019 Ngày nộp đề tài: 09/12/2019

1. Tên đề tài: Xây dựng website quản lý Spa

2. Các số liệu, tài liệu ban đầu: Không

1. Nội dung thực hiện đề tài:

Tìm hiểu lý thuyết về các công nghệ, thuật toán áp dụng để thực hiện đề tài

* + - Font-end: HTML5, CSS3, Javascript
    - Back-end: ASP.NET MVC 5
    - Database: hệ quản trị cơ sở dữ liệu SQL Sever
    - Tìm hiểu lý thuyết về hệ khuyến nghị
    - Tìm hiểu các thuật toán áp dụng cho hệ khuyến nghị

1. Sản phẩm: Ứng dụng website quản lý Spa.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRƯỞNG NGÀNH  Nguyễn Đăng Quang | | | GIẢNG VIÊN HƯỚNG DẪN  Nguyễn Thiên Bảo | |
|  | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  **\*\*\*\*\*\*\*** | |

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên: Lê Minh Phước MSSV: 15110103

Họ và tên Sinh viên: Vũ Ngọc Phát MSSV: 15110096  
Ngành: Công nghệ thông tin  
Tên đề tài: Xây dựng website quản lý Spa   
Họ và tên Giảng viên hướng dẫn: TS. Nguyễn Thiên Bảo

NHẬN XÉT1. Về nội dung đề tài & khối lượng thưc hiện:

Nhóm đã hoàn thành được các mục tiêu đề ra ban đầu của đề tài được giao trong khoảng thời gian xác định.

- Về lý thuyết:

+ Nắm được kiến thức về ASP.NET MVC 5, HTML, CSS, Javascript, SQL Sever

+ Sinh viên nắm được kiến thức mô hình quản lý Spa

- Về thực hành:

+ Sinh viên hoàn thành được website quản lý Spa

2. Ưu điểm:

- Sinh viên chịu khó tìm hiểu, tìm kiếm các tài liệu liên quan để thực hiện đề tài

- Sinh viên thực hiện các công việc tốt và đúng deadline được giao

- Thái độ làm việc:làm việc nghiêm túc, hoàn thành các công việc được giao, có trách nhiệm với công việc, chủ động liên hệ với GVHD

3. Khuyết điểm:

- Cần cải thiện thêm UI/UX

- Chức năng còn cần hoàn thiện thêm

4. Đánh giá loại:

5. Điểm: (Bằng chữ: )

*Tp. Hồ Chí Minh, ngày tháng năm 20…..*

Giảng viên hướng dẫn

*(Ký & ghi rõ họ tên)*

|  |  |
| --- | --- |
|  | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  **\*\*\*\*\*\*\*** |

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên: Lê Minh Phước MSSV: 15110103

Họ và tên Sinh viên: Vũ Ngọc Phát MSSV: 15110096  
Ngành: Công nghệ thông tin  
Tên đề tài: Xây dựng website quản lý Spa  
Họ và tên Giảng viên phản biện:

NHẬN XÉT1. Về nội dung đề tài & khối lượng thưc hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đánh giá loại:

5. Điểm: (Bằng chữ: )

*Tp. Hồ Chí Minh, ngày tháng năm 20…..*

Giảng viên phản biện

*(Ký & ghi rõ họ tên)*

**LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Nay em xin được phép gửi lời cảm ơn chân thành này đến thầy Nguyễn Thiên Bảo, người đã trực tiếp hỗ trợ em trong suốt quá trình định hướng chọn đề tài, hướng dẫn, nhận xét và góp ý cũng như cung cấp tài liệu tham khảo. Nếu không có những lời hướng dẫn, những kinh nghiệm thực tiễn của thầy thì em nghĩ rằng đồ án này sẽ khó có thể hoàn thiện và hoàn thành đúng thời hạn được. Một lần nữa, em xin cảm ơn thầy.

Em cũng xin chân thành cảm ơn các quý thầy cô trong khoa Đào tạo Chất lượng cao đã giúp đỡ hỗ trợ kiến thức cũng như giải đáp thắc mắc của em. Cùng với đó, em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp cho em hoàn thiện đề tài hơn.

Khoảng thời gian có hạn, cùng với kiến thức còn hạn chế và còn nhiều bỡ ngỡ khác do đó thiếu sót là điều không thể tránh khỏi nên em rất mong nhận được những ý kiến đóng góp quý báo từ các quý Thầy Cô để kiến thức của em được hoàn thiện hơn sau này. Em xin chân thành cảm ơn.

Thành phố Hồ Chí Minh, ngày 7 tháng 12 năm 2019

**Sinh viên thực hiện**

Lê Minh Phước Vũ Ngọc Phát

**TÓM TẮT**

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang đi vào cao trào thì vai trò của website ngày càng quan trọng. Thậm chí một số doanh nghiệp phát triển đến 80 – 90% doanh thu từ website. Website không những đã trở nên rất phổ biến mà còn là công cụ không thể thiếu đối với mỗi doanh nghiệp và cá nhân hoạt động kinh doanh hiện nay. Đây là phương thức quảng bá thông tin nhanh nhạy và tiếp cận khá hiệu quả đối với người dùng Internet. Website được đánh giá là công cụ dễ dàng để chăm sóc khách hàng tiềm năng đồng thời mở rộng khách hàng mục tiêu đối với mỗi doanh nghiệp. Mặt khác, website chính là bước tiến quan trọng và hiệu quả trong việc thực hiện chiến lược marketing của công ty. Đối với cá nhân, sử dụng các mạng xã hội, xem thông tin trên website là cách đơn giản và nhanh nhất để có thể nắm được đầy đủ các thông tin cần thiết của sản phẩm, dịch vụ và là cách đơn giản để tìm kiếm khách hàng, thu thập thông tin của các đối tác quan trọng cho công việc kinh doanh.

**MỤC LỤC**

[CHƯƠNG I: MỞ ĐẦU 4](#_Toc26745438)

[1. Lý do chọn đề tài 4](#_Toc26745439)

[2. Mục tiêu đề tài 5](#_Toc26745440)

[3. Nội dung 5](#_Toc26745441)

[4. Phương pháp nghiên cứu 5](#_Toc26745442)

[5. Ý nghĩa khoa học thực tiễn 5](#_Toc26745443)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 5](#_Toc26745444)

[1. ASP.NET MVC 5](#_Toc26745445)

[1.1. Giới thiệu về ASP.NET MVC 6](#_Toc26745446)

[1.1.1. Lịch sử 6](#_Toc26745447)

[1.1.2. Giới thiệu: 7](#_Toc26745448)

[1.1.3. Xây dựng theo mô hình MVC: 7](#_Toc26745449)

[1.1.4. Lợi ích khi xây dựng ứng dụng web dựa trên mô hình MVC 9](#_Toc26745450)

[1.1.5. Tính năng của ASP.Net MVC 9](#_Toc26745451)

[1.2. Kiến trúc ứng dụng Asp.Net MVC 10](#_Toc26745452)

[1.2.1. Giao tiếp trong kiến trúc Asp.Net MVC 10](#_Toc26745453)

[1.2.2. Các vấn đề cơ bản về Url và Routing 11](#_Toc26745454)

[1.3. Những ưu điểm mà ASP.Net MVC mạng lại cho chúng ta 11](#_Toc26745455)

[2. SQL SEVER 12](#_Toc26745456)

[2.1. Giới thiệu về Sql Sever 12](#_Toc26745457)

[2.1.1. Lịch sử ra đời và một vài bản ấn: 12](#_Toc26745458)

[2.1.2. Các thành phần cơ bản trong SQL Server 13](#_Toc26745459)

[2.2. Tại sao lại sử dụng SQL trong thiết kế Web 13](#_Toc26745460)

[CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ ỨNG DỤNG 15](#_Toc26745461)

[1. Phân tích yêu cầu 15](#_Toc26745462)

[1.1. Nghiệp vụ 15](#_Toc26745463)

[1.1.1. Đối tượng sử dụng 15](#_Toc26745464)

[1.1.2. Chức năng 15](#_Toc26745465)

[2. Thiết kế xử lý 17](#_Toc26745466)

[2.1. Use case 17](#_Toc26745467)

[2.2. Đặc tả Use case 17](#_Toc26745468)

[2.2.1. Use case Đăng nhập 17](#_Toc26745469)

[2.2.2. Use case Chỉnh sửa thông tin tài khoản 18](#_Toc26745470)

[2.2.3. Use case Quản lý dịch vụ 19](#_Toc26745471)

[2.2.4. Use case Quản lý tài khoản 19](#_Toc26745472)

[2.3. Class Diagram 20](#_Toc26745473)

[2.3.1. Lớp Controller 20](#_Toc26745474)

[2.3.2. Lớp Model 21](#_Toc26745475)

[2.3.3. Lớp View 22](#_Toc26745476)

[2.4. Thiết kế cơ sở dữ liệu 22](#_Toc26745477)

[2.4.1. Sơ đồ ERD 22](#_Toc26745478)

[2.4.2. Danh sách chi tiết thuộc tính các bảng 23](#_Toc26745479)

[2.5. Thiết kế giao diện: 31](#_Toc26745480)

[2.5.1. Danh sách các màn hình: 31](#_Toc26745481)

[2.5.2. Đặc tả các màn hình 31](#_Toc26745482)

[2.6. Hiện thực hệ thống 39](#_Toc26745483)

[2.6.1. Công nghệ sử dụng 39](#_Toc26745484)

[2.6.2. Môi trường lập trình 39](#_Toc26745485)

[CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 40](#_Toc26745486)

[1. Tổng kết 40](#_Toc26745487)

[1.1. Về mặt lý thuyết 40](#_Toc26745488)

[1.2. Về mặt hiện thực 40](#_Toc26745489)

[2. Hạn chế 41](#_Toc26745490)

[3. Hướng phát triển đề tài: 41](#_Toc26745491)

CHƯƠNG I: MỞ ĐẦU

1. **Lý do chọn đề tài**
2. **Mục tiêu đề tài**

Xây dựng website quản lý spa

1. **Nội dung**

Tìm hiểu lý thuyết về các công nghệ, thuật toán áp dụng để thực hiện đề tài

* ASP.NET MVC 5
* Tìm hiểu về quản lý cơ sở dữ liệu trên SQL Sever

Phân tích yêu cầu và hiện thực hóa hệ thống

* Phân tích yêu cầu
* Mô hình hóa hệ thống
* Thiết kế xử lí

1. **Phương pháp nghiên cứu**

Phương pháp tổng hợp lý thuyết:

* Nghiên cứu và tìm hiểu các tài liệu, website liên quan đến các công nghệ đang tìm hiểu.

Phương pháp chuyên gia:

* Tham khảo ý kiến của Thầy hướng dẫn cùng các anh, chị và bạn bè có kinh nghiệm liên quan đến các vấn đề của khóa luận để tạo tính chính xác và khoa học của đề tài.

Phương pháp mô hình hóa:

* Mô phỏng ứng dụng từ bước thiết kế cài đặt cho đến kết quả thành phẩm của ứng dụng.

1. **Ý nghĩa khoa học thực tiễn**

Website có giao diện trực quan, dễ sử dụng, thoải mái, thuận tiện, giúp dễ dàng tìm kiếm các quán ăn phù hợp, hợp lý cho người dùng.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

1. **ASP.NET MVC**

## 1.1. Giới thiệu về ASP.NET MVC

### **1.1.1. Lịch sử**

Asp.Net MVC là một Framework sử dụng .Net Framework cho việc phát triển ứng dụng web động, trước khi Asp.Net MVC ra đời, lập trình viên sử dụng công nghệ Asp.Net Web Form trên nền tảng .Net Framework để phát triển ứng dụng Web động.

Lịch sử phát triển của ASP.NET MVC:

*Asp.Net 1.0 :* Phiên bản đầu tiên được phát hành vào tháng 3 năm 2009 trên nền tảng bộ .Net Framework 3.5 bộ Visual Studio 2008.  
*Asp.Net 2.0 :* Phiên bản này được phát hành vào tháng 3 năm 2010 trên bộ .Net Framework 3.5 và .Net Framework 4.0 bộ Visual Studio 2010.   
*Asp.Net MVC 3 :* Phiên bản này được phát hành vào tháng 1 năm 2011 trên nền tảng bộ .Net Framework 4.0, các đặc điểm chính của phiển bản này :

* So với phiên bản ASP.NET MVC2 thì tính năng mới được hỗ trợ thêm HTML5 và CSS3; tính năng này chỉ hỗ trợ cho những trình duyệt mới;
* Cải thiện về Model validation;
* Hỗ trợ 2 View Engine: RAZOR, ASPX, kể cả Open source trong khi ASP.NET MVC2 chỉ có ASPX;
* Controller được cải tiến hơn như thuộc tính ViewBag và kiểu ActionResult;
* Cải thiện Dependency Injection với IDpendencyResolver (có 2 phần: DependencyResolver và interface IDpendencyResolver)trong ASP.NET MVC3; đây là lớp thực thi mô hình Service Locator, cho phép framework gọi DIContainer khi cần làm việc với 1 lớp thực thi từ 1 kiểu cụ thể
* Cách tiếp cận với JavaScript được hạn chế;
* Hỗ trợ caching trong Partial page;

*Asp.Net 4.0 :* Phiên bản này được phát hành vào tháng 8 năm 2012 trên nền tảng bộ .Net Framework 4.0 và .Net Framework 4.5 trên phiên bản Visual Studio 2010 SP1 and Visual Studio 2012. Phiên bản này có các đặc điểm chính như sau :

* ASP.NET Web API ra đời, nhằm đơn giản hoá việc lập trình với HTML hiện đại và đây là một cải tiến mới thay cho WCF Web API;
* Mặc định của dự án được cải thiện hơn về hình thức bố trí, giúp dễ nhìn hơn;
* Mẫu Empty Project là project trống, phù hợp cho những tín đồ developer muốn nâng cao khả năng lập trình với ASP.NET MVC4;
* Giới thiệu jQuery Mobile, và mẫu Mobile Project cho dự án;
* Hỗ trợ Asynchrnous Controller;
* Kiểm soát Bundling và Minification thông qua web.config
* Hỗ trợ cho việc đăng nhập OAuth và OpenID bằng cách sử dụng thư viện DotNetOpenAuth;
* Phiên bản mới Windows Azure SDK 1.6 được phát hành;

*Asp.Net MVC 5 :* Phiên bản này được phát hành vào tháng 10 năm 2013 trên nền tảng bộ .Net Framework 4.5 và .Net Framework 4.5.1 trên phiên bản Visual Studio 2013. Các đặc điểm chính của phiên bản này như sau :

* Với MVC5 thì cải tiến hơn so với ASP.NET MVC4, Bootstrap được thay thế mẫu MVC mặc định;
* Chứng thực người dùng Authentication Filter được tuỳ chỉnh hoặc chứng thực từ hãng thứ 3 cung cấp;
* Với Filter overrides, chúng ta có thể Filter override trên Method hoặc Controller;
* Thuộc tính Routing được tích hợp vào MVC5;

### **1.1.2. Giới thiệu:**

Asp.Net MVC là một Framework sử dụng .Net Framework cho việc phát triển ứng dụng web động, trước khi Asp.Net MVC ra đời, lập trình viên sử dụng công nghệ Asp.Net Web Form trên nền tảng .Net Framework để phát triển ứng dụng Web động.

Phiên bản Asp.Net MVC 5 được phát hành vào tháng 10 năm 2013 trên nền tảng bộ .Net Framework 4.5 và .Net Framework 4.5.1 trên phiên bản Visual Studio 2013.

### **1.1.3. Xây dựng theo mô hình MVC:**

Asp.Net MVC phát triển trên mẫu thiết kế chuẩn MVC, cho phép người sử dụng phát triển các ứng dụng phần mềm. **MVC** là tên một mẫu phát triển ứng dụng, phương pháp này chia nhỏ một ứng dụng thành ba thành phần để cài đặt, mỗi thành phần đóng một vai trò khác nhau và ảnh hưởng lẫn nhau, đó là models, views, và controllers.



*Hình 1.1: Mẫu thiết kế MVC*

*Models*: Các đối tượng Models là một phần của ứng dụng, các đối tượng này 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 như, một đối tượng Employee (nhân viên) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Eployees ở SQL Server. Trong các ứng dụng nhỏ, model thường là chỉ là một khái niệm nhằm phân biệt hơn là được cài đặt thực thụ, ví dụ, nếu ứng dụng chỉ đọc dữ liệu từ CSDL và gởi chúng đến view, ứng dụng khong cần phải có tầng model và các lớp lien quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model).  
  
*Views:* 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 Employees sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Employee.  
  
*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.

### **1.1.4. Lợi ích khi xây dựng ứng dụng web dựa trên mô hình MVC**

- Nền tảng ASP.NET MVC mang lại những lợi ích sau:

* Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller
* Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD)
* Hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng.

### **1.1.5.** **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 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 MVC sử dụng view engine 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.

## 1.2. Kiến trúc ứng dụng Asp.Net MVC

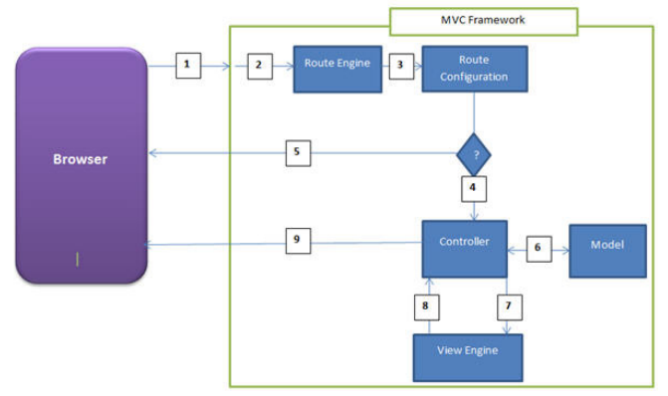
### **1.2.1. Giao tiếp trong kiến trúc Asp.Net MVC**

Các thành phần cơ bản của ứng dụng Asp.Net MVC bao gồm :

* MVC Framework
* Route engine
* Route configuration
* Control
* Model
* View engine
* View

Các thành phần này giao tiếp với nhau nhằm xử lý các yêu cầu của ứng dụng Asp.Net MVC. Quá trình xử lý yêu cầu thường liên quan tới một chuỗi các xử lý, mỗi xử lý sẽ được một component trong Asp.net đảm nhiệm.

1. Trình duyệt gửi yêu cầu của ứng dụng Asp.Net MVC
2. MVC Engine chuyển yêu cầu tới cho Routing engine (Bộ điều hướng)
3. Routing engine kiểm tra cấu hình điều hướng (route configuration) của ứng dụng nhằm chuyển đến các controller phù hợp với yêu cầu.
4. Khi Controller được tìm thấy, control này sẽ được thực thi
5. Nếu Controller không được tìm thấy, bộ điều hướng sẽ chỉ ra rằng controller không được tìm thấy và MVC Engine sẽ thông báo lỗi cho trình duyệt.
6. Controller giao tiếp với model, tầng Model đại diện cho các Entity bên trong hệ thống, Controller làm việc với tầng model để lấy thông tin cần cho người dùng.
7. Controller yêu cầu View engine hiển thị thông tin dựa trên dữ liệu của tầng model
8. View engine trả kết quả cho controller
9. Controller gửi kết quả cho trình duyệt thông qua giao thức Http.



*Hình 1.2:* Giao tiếp trong kiến trúc Asp.Net MVC

### **1.2.2. Các vấn đề cơ bản về Url và Routing**

Hầu hết các Web Framework như Asp.net Web Form, jsp … đều ánh xạ url vào 1 file được lưu trữ trên ổ đĩa.  
*Ví dụ :*

Địa chỉ Url : <http://abc.com/default.aspx>, quá trình thực hiện sẽ điều hướng tới file default.aspx trên ổ đĩa vật lý ví dụ D:\webroot\default.aspx

Asp.net MVC làm việc theo cách khác, thay vì ánh xạ các file trên ổ đĩa, nó sẽ trực tiếp ánh xạ vào các lớp, đó chính là các controller.  
*Ví dụ:*  
Địa chỉ Url : <http://abc.com/>admin, quá trình thực hiện sẽ điều hướng controller là admin, Action method mặc định trong controller là Index sẽ được gọi.

## 1.3. Những ưu điểm mà ASP.Net MVC mạng lại cho chúng ta

* Do sử dụng mô hình MVC nên trong ASP.Net MVC đã tách biệt được các tầng trong mô hình lập trình web vì vậy giúp tối ưu ứng dụng và dễ dàng trong việc viết code, giao diện
* Giao diện trong ASP.Net MVCsử dụng công nghệ thiết kế web HTML, CSS nền việc thiết kế giao diện trở nên dễ dàng và giúp cho designer linh hoạt trong việc thiết kế
* ASP.Net MVC không sử dụng view state vì vậy trang web không bị tăng kích thước do đó hiệu năng hoạt động không bị bị giảm

ASP.Net MVC đã khắc phục được các nhược điểm của web forms vì vậy web forms hiện nay không còn được dùng phổ biến nữa.

# **2. SQL SEVER**

## 2.1. Giới thiệu về Sql Sever

SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, datase engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

### **2.1.1. Lịch sử ra đời và một vài bản ấn:**

Lịch sử ra đời:

* Năm 1989, phiên bản đầu tiên của SQL Server 1.0 ra đời được dùng cho các hệ điều hành 16 bit và được phát triển cho tới ngày nay.
* Cho tới khi SQL Server ra phiên bản 6.5 thì được thị trường chấp nhận rộng rãi. Một đột phá cải tiến cho SQL Server 7.0 khi được Microsoft viết lại một engine hoàn toàn mới. Đến khi SQL Server từ phiên bản 7.0 cải tiến lên 8.0 chủ yếu phát triển về tính năng thiết kế web.
* Cho đến ngày nay thì phiên bản mới nhất đó là SQL Server 2019 hỗ trợ bộ vi xử lý 64 bit ra đời vào ngày 4 tháng 11 năm 2019.

Một vài bản ấn:

* Enterprise: là một ấn bản chứa tất cả các đặc điểm nổ bật của SQL Server như: các công cụ cho tạo và quản lý phân cụm SQL Server, nhân bộ máy cơ sở dữ liệu và một số dịch vụ đi kèm. Nó có thể đánh địa chỉ 12 terabytes và quản lý cơ sở dữ liệu lên tới 524 petabytes.
* Standard: Ấn bản này có thể chạy tốt trên hệ thống lên tới 4 CPU và 2 GB RAM rất thích hợp cho các dịch vụ thiết kế web vừa và nhỏ.
* Developer: Ấn bản này giới hạn số lượng người kết nối với server nhưng có đầy đủ các tính năng của Enterprise Edition. Đây là phiên bản được sử dụng cho kiểm tra và phát triển ứng dụng phù hợp cho các cá nhân trong lĩnh vực web như: [freelancer Việt Nam](http://freelancervietnam.vn/).
* Workgroup: ấn bản SQL Server này có các chức năng lõi cơ sở dữ liệu nhưng không đi kèm các dịch vụ. Ở phiên bản 2012 không có ấn bản này.
* Express: Ấn bản này dễ dàng sử dụng và quản trị cơ sở dữ liệu đơn giản.

**2.1.2. Các thành phần cơ bản trong SQL Server**

Các thành cơ bản trong SQL Server gồm có: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service,… Tất cả kết hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.

**+ Database Engine**: Đây là một engine có khả năng chứa dữ liệu ở các quy mô dưới dạng support và table. Ngoài ra, nó còn có khả năng tự điều chỉnh ví dụ: trả lại tài nguyên cho ệ điều hành khi một user log off và sử dụng thêm các tài nguyên của máy khi cần.

**+ Integration Services**: là tập hợp các đối tượng lập trình và các công cụ đồ họa cho việc sao chép, di chuyển và chuyển đổi dữ liệu.  Khi bạn làm việc trong một công ty lớn thì dữ liệu được lưu trữ ở nhiều nơi khác nhau như được chứa trong: Oracle, SQL Server, DB2, Microsoft Access,… và bạn chắc chắn sẽ có nhu cầu di chuyển dữ liệu giữa các server này. Ngoài ra, bạn còn muốn định dạng dữ liệu trước khi lưu vào database. Chắc chắn Integration Services sẽ giúp bạn giải quyết được công việc này dễ dàng.

**+ Analysis Services**: Đây là một dịch vụ phân tích dữ liệu rất hay của Microsoft. Dữ liệu khi được lưu trữ vào trong database mà bạn không thể lấy được những thông tin bổ ích thì coi như không có ý nghĩa gì. Chính vì thế, công cụ này ra đời giúp bạn trong việc phân tích dữ liệu một cách hiệu quả và dễ dàng bằng cách dùng kỹ thuật khai thác dữ liệu – datamining và khái niệm hình khối nhiều chiều – multi dimendion cubes.

**+ Notification Services**: Dịch vụ thông báo này là nền tảng cho sự phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này còn có chức năng gửi thông báo theo dịch thời đến hàng ngàn người dăng ký sử dụng trên nhiều loại thiết bị khác nhau.

**+ Reporting  Services**: là một công cụ tạo, quản lý và triển khai báo cáo bao gồm: server và client. Ngoài ra, nó còn là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.

**+ Full Text Search Service**: là một thành phần đặc biệt trong việc truy vấn và đánh chỉ mục dữ liệu văn bản không cấu trúc được lưu trữ trong các cơ sở dữ liệu SQL Server.

**+ Service Broker**: là một môi trường lập trình cho việc tạo ra các ứng dụng trong việc nhảy qua các Instance.

**2.2. Tại sao lại sử dụng SQL trong thiết kế Web**

QL Server không phải là một hệ quản trị cơ sở dữ liệu độc lập mà nó chỉ là một thành phần với vai trò ngôn ngữ là công cụ giao tiếp giữa hệ cơ sở dữ liệu và người dùng. Chính vì thế nó được sử dụng trong các dịch vụ thiết kế web đẹp với chức năng giao tiếp với người dùng với các vai trò sau:

SQL là một ngôn ngữ đòi hỏi có tính tương tác cao: Người dùng có thể dễ dàng trao đổi với các tiện ích thông qua các câu lệnh của SQL đến cơ sở dữ liệu và nhận kết quả từ cơ sở dữ liệu.

SQL là một ngôn ngữ lập trình cơ sở dữ liệu: Các lập trình viên có thể xây dựng các chương trình ứng dụng giao tiếp với cơ sở dữ liệu bằng cách nhúng các câu lệnh SQL vào trong ngôn ngữ lập trình.

SQL là một ngôn ngữ lập trình quản trị cơ sở dữ liệu: Người quản trị cơ sở dữ liệu có thề quản lý, định nghĩa và điều khiển truy cập cơ sở dữ liệu thông qua SQL.

SQL là một ngôn ngữ lập trình cho các hệ thống chủ khách: SQL được sử dụng như là một công cụ giao tiếp với các trình ứng dụng trong hệ thống cơ sở dữ liệu khách chủ.

SQL là ngôn ngữ truy cập dữ liệu trên Internet: SQL được sử dụng với vai trò tương tác với dữ liệu trong hầu hết các máy chủ web và máy chủ Internet.

SQL là ngôn ngữ cơ sở dữ liệu phân tán: Với vai trò giao tiếp với các hệ thống trên mạng, gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.

CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ ỨNG DỤNG

# **1. Phân tích yêu cầu**

## 1.1. Nghiệp vụ

### **1.1.1. Đối tượng sử dụng**

- Những ai có nhu cầu tìm kiếm dịch vụ spa. (End - User)

- Người quản trị. (Admin)

### **1.1.2. Chức năng**

- Người quản trị. (Admin)

|  |  |  |
| --- | --- | --- |
| STT | Trang quản trị nội dung | Tính năng |
| 1 | Trang chủ | Giao diện trang chủ quản trị |
| 2 | Dash Board | Hiển thị thông tin Visitor, Customer, Booking, Services |
| 3 | Dash Board | Hiển thị Chart theo thông tin Visitor, Customer, Booking, Services |
| 4 | Dash Board | Hiển thị lịch |
| 6 | Booking | Xem, Thêm, xóa , sửa Booking |
| 7 | Booking | Thêm Booking: Hiển thị spa bed còn trống theo thời gian khách hàng đến |
| 8 | Booking | Hiển thị schedule xem phòng nào đang có người booking, theo giờ, ngày |
| 10 | Product | Xem, Thêm, xóa, sửa Product |
| 9 | Product Categories | Xem danh sách Product Categories |
| 12 | Services | Xem, Thêm, xóa, sửa Services |
| 12 | Services Categories | Xem, Thêm, xóa, sửa Services Categories |
| 14 | Blog | Xem, Thêm, xóa, sửa Blog |
| 15 | Blog Categories | Xem, Thêm, xóa, sửa Blog Categories |
| 17 | Room | Xem, thêm, xóa, sửa room |
| 17 | Bed | Xem, thêm, xóa, sửa bed |
| 17 | Customer | Xem, xóa, sửa Customer |
| 17 | About | Xem, thêm, xóa, sửa Our Story, Testimonial |
| 17 | Footer | Xem, thêm, xóa, sửa Footer, Link mạng xã hội,.. |
| 17 | Footer Categories | Xem, thêm, xóa, sửa Footer Categories |
| 17 | Menu | Xem, thêm, xóa, sửa Menu |
| 17 | Slide | Xem, thêm, xóa, sửa Slide |

*Bảng 2.1. Tính năng của trang quản trị nội dung người quản trị*

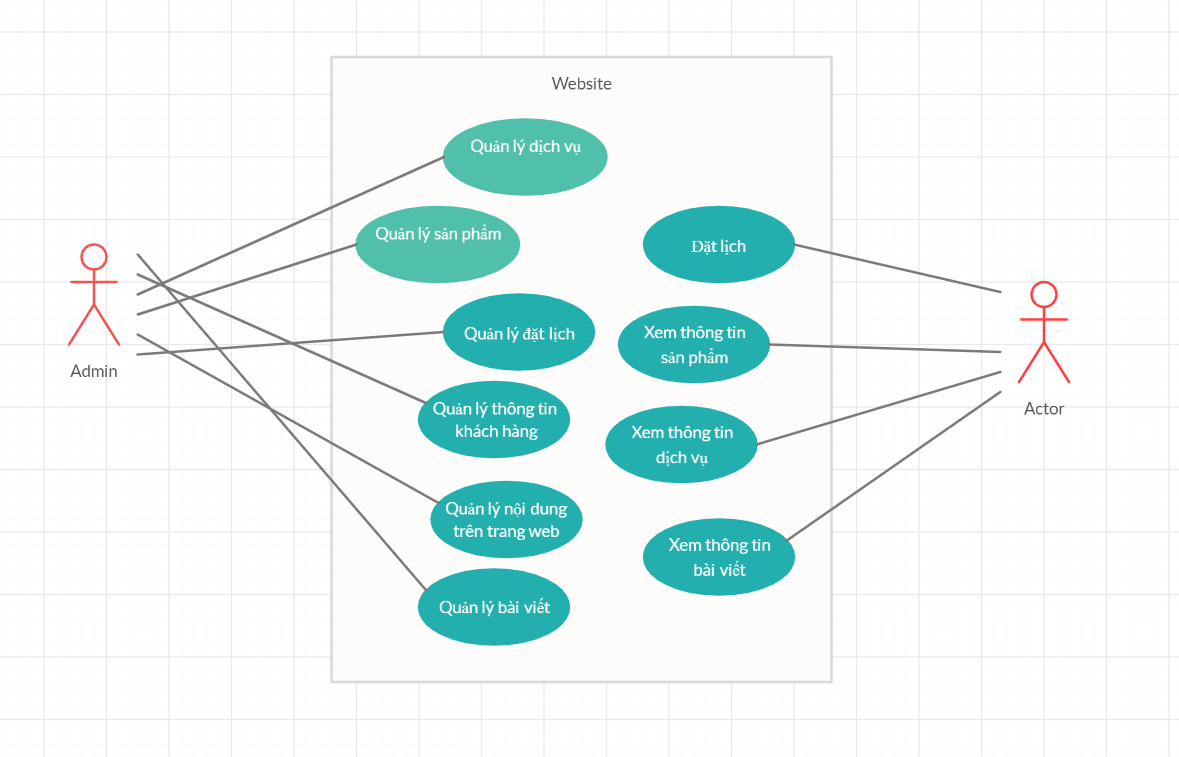
- End-User

|  |  |  |
| --- | --- | --- |
| STT | Trang quản trị nội dung | Tính năng |
| 1 | Trang chủ | Trang chủ |
| 2 | Trang chủ => Slide | Hiển thị Slide, có Link đến bài viết hoặc sản phẩm, có chức năng tự chạy |
| 3 | Trang chủ => Our Story | Hiển thị Our Story trên trang chủ |
| 4 | Trang chủ => Featured Services | Hiển thị Featured Services trên trang chủ |
| 5 | Trang chủ =>Booking | Booking trên trang chủ |
| 6 | Services | Hiển thị danh sách services |
| 7 | Services | Xem chi tiết services |
| 8 | Services | Phân trang danh sách services |
| 9 | Products | Hiển thị danh sách Product |
| 10 | Products | Xem chi tiết Product |
| 11 | Products | Phân trang danh sách Product |
| 9 | Booking | Hiển thị danh sách Room - Bed, Time để khách hàng booking |
| 10 | Booking | Booking thành công => Trang success thông báo ID Booking |
| 11 | About us | Hiển thị Our Story, List nhân viên, List Testimonial |
| 11 | Contact | Hiển thị thông tin contact, Map của Spa |

*Bảng 2.2. Tính năng của trang quản trị nội dung khách hàng*

# **2. Thiết kế xử lý**

## 2.1. Use case



*Hình 2.1. Lược đồ chức năng Use case*

## 2.2. Đặc tả Use case

### **2.2.1. Use case Đăng nhập**

|  |  |
| --- | --- |
| **Name** | Đăng nhập |
| **Brief description** | Người quản trị đăng nhập vào hệ thống |
| **Actor(s)** | Người quản trị |
| **Pre-conditions** | Người quản trị đã có tài khoản trong hệ thống, có kết nối Internet |
| **Post-conditions** | Nếu đăng nhập thành công: Người dùng được xác thực và hệ thống hiển thị thông tin người dùng  Nếu đăng nhập không thành công: Người dùng không đăng nhập được vào hệ thống |
| **Flow of events** |  |
| **Basic flow**  **(Thành công)** | Use case bắt đầu khi người dùng click vào ô đăng nhập.   1. Hệ thống hiển thị cửa sổ cho người dùng nhập username và password 2. Người dùng nhập username và password 3. Hệ thống kiểm tra trong cơ sở dữ liệu xác thực username và password 4. Hệ thống mở màn hình trang chủ và mở các chức năng được phép truy cập của người dùng. |
| **Alternative flow (Thất bại)** | Người dùng nhập sai username hoặc password  => Hệ thống thông báo lỗi đăng nhập yêu cầu đăng nhập lại |
| **Extension point** | Hệ thống thông báo lỗi |

### **2.2.2. Use case Chỉnh sửa thông tin tài khoản**

|  |  |
| --- | --- |
| **Name** | Chỉnh sửa thông tin tài khoản |
| **Brief description** | Người dùng có thể thay đổi thông tin cá nhân của mình |
| **Actor(s)** | Người quản trị |
| **Pre-conditions** | Có kết nối Internet |
| **Post-conditions** |  |
| **Flow of events** |  |
| **Basic flow**  **(Thành công)** | Use case bắt đầu khi người quản trị đang ở trang Admin và click lên [Sửa]   1. Hệ thống hiển thị trang chỉnh sửa thông tin 2. Người dùng tiến hành thay đổi thông tin cần sửa 3. Bấm Save 4. Hệ thống lưu thông tin chỉnh sửa xuống cơ sở dữ liệu 5. Thông tin của người quản trị được thay đổi 6. Quay về màn hình trước |
| **Alternative flow (Thất bại)** | Tại bước 3 khi người dùng nhấn cancel.  => Quay về màn hình trước đó |
| **Extension point** | Không có |

### **2.2.3.** **Use case Quản lý dịch vụ**

|  |  |
| --- | --- |
| **Name** | Quản lí dịch vụ |
| **Brief description** | Thêm, sửa, xóa thông tin dịch vụ |
| **Actor(s)** | Người quản trị |
| **Pre-conditions** | Có kết nối Internet |
| **Post-conditions** | Thành công: Thông tin dịch vụ thay đổi theo người quản trị  Không thành công: Dữ liệu dịch vụ không thay đổi |
| **Flow of events** |  |
| **Basic flow**  **(Thành công)** | Use case bắt đầu khi người dùng click vào các biểu tượng thêm , sửa, xóa. Nhập theo yêu cầu  Chọn Lưu |
| **Alternative flow (Thất bại)** | Người quản trị chọn Hủy |
| **Extension point** | Không có |

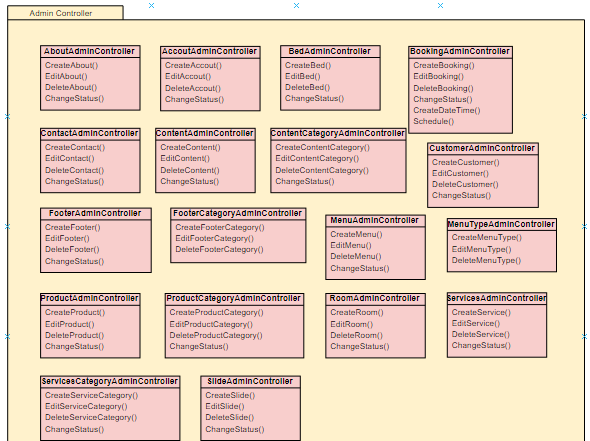
### **2.2.4. Use case Quản lý tài khoản**

|  |  |
| --- | --- |
| **Name** | Quản lí tài khoản |
| **Brief description** | Xem và xóa thông tin tài khoản |
| **Actor(s)** | Người quản trị |
| **Pre-conditions** | Có kết nối Internet |
| **Post-conditions** | Thành công: Thông tin của tài khoản thay đổi theo người quản trị  Không thành công: Thông tin tài khoản không thay đổi |
| **Flow of events** |  |
| **Basic flow**  **(Thành công)** | Use case bắt đầu khi người dùng click vào các biểu tượng thêm , sửa, xóa  Nhập theo yêu cầu  Chọn Lưu |
| **Alternative flow (Thất bại)** | Người quản trị chọn Hủy |
| **Extension point** | Không có |

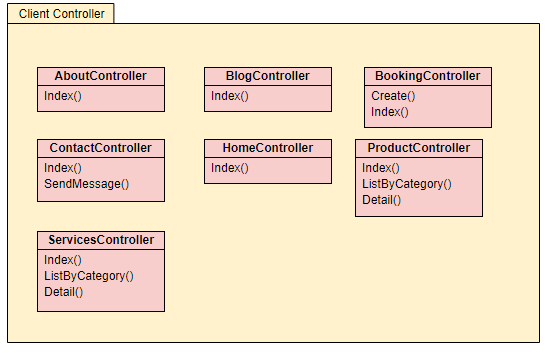
## 2.3. Class Diagram

### **2.3.1. Lớp Controller**

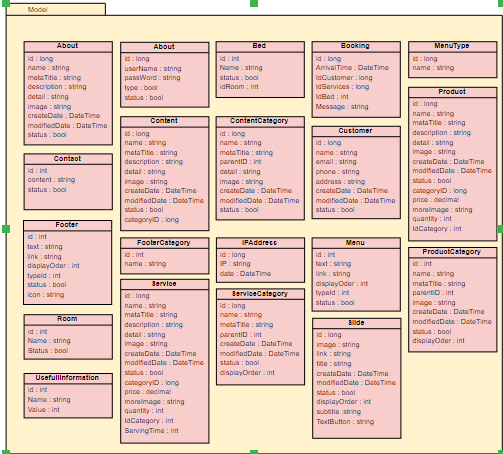
**Admin Controller**



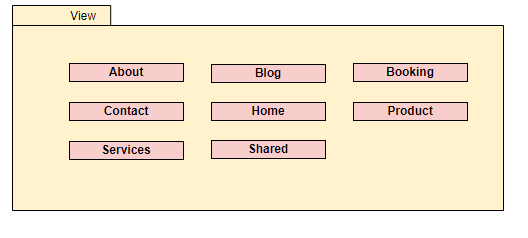
**Client Controller**



### **2.3.2. Lớp Model**

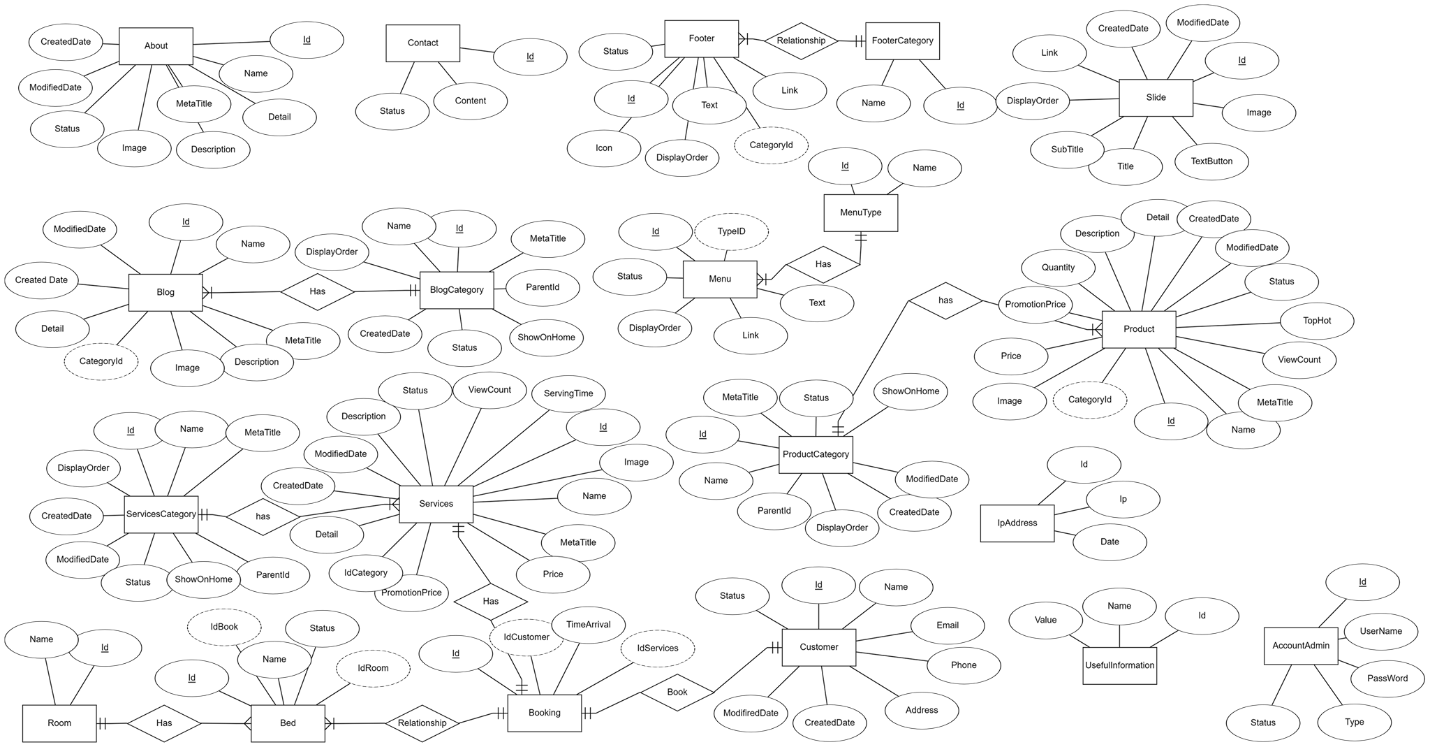
****

### **2.3.3. Lớp View**



## 2.4. Thiết kế cơ sở dữ liệu

### **2.4.1. Sơ đồ ERD**

****

|  |  |
| --- | --- |
| **Tên bảng** | **Mô tả** |
| About | Bảng thông tin về Spa |
| Bed | Bảng thông tin về Bed |
| Booking | Bảng thông tin đặt lịch (Booking) |
| Contact | Bảng thông tin liên hệ |
| Content | Bảng thông tin về bài viết trên trang Spa |
| ContentCategory | Bảng thông tin danh mục bài viết |
| Customer | Bảng thông tin về khách hàng |
| AccountAdmin | Bảng thông tin về tài khoản người quản trị |
| Footer | Bảng thông tin Footer |
| FooterCategory | Bảng thông tin danh mục Footer |
| IPAdress | Bảng thông tin về địa chỉ IP của người truy cập trang web |
| Menu | Bảng thông tin Menu |
| MenuType | Bảng thông tin về loại Menu |
| Product | Bảng thông tin về sản phẩm |
| ProductCategory | Bảng thông tin về danh mục sản phẩm |
| Room | Bảng thông tin về phòng |
| Service | Bảng thông tin về các dịch vụ |
| ServiceCategory | Bảng danh mục về các dịch vụ |
| Slide | Bảng slide |
| UsefullInformation | Bảng thông tin thống kê mà người quản trị cần |

### **2.4.2. Danh sách chi tiết thuộc tính các bảng**

Bảng “About”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã About (khóa chính) |
| 2 | name | nvarchar | Tên About |
| 3 | metaTitle | nvarchar | Tên tiêu đề |
| 4 | description | nvarchar | Mô tả |
| 5 | createDate | date | Ngày tạo |
| 6 | modifiedDate | date | Ngày chỉnh sửa |
| 7 | status | bit | Trạng thái |

Bảng “Bed”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã Bed |
| 2 | name | nvarchar | Tên About |
| 3 | idRoom | int | Mã phòng |
| 4 | status | bit | Trạng thái |

Bảng “Booking“

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã Booking (khóa chính) |
| 2 | ArrivalTime | datetime | Thời gian đặt lịch |
| 3 | IdCustomer | int | Mã khách hàng |
| 4 | IdServices | int | Mã dịch vụ |
| 5 | IdBed | int | Mã Bed |
| 6 | Message | nvachar | Tin nhắn khách hàng nhắn gửi |

Bảng “Contact”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã Contact |
| 2 | content | ntext | Tên thông tin liên hệ |
| 4 | status | bit | Trạng thái |

Bảng “Content”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã Content (khóa chính) |
| 2 | name | nvarchar | Tên Content |
| 3 | metaTitle | nvarchar | Tên tiêu đề |
| 4 | image | nvarchar | Hình ảnh |
| 5 | categoryID | int | Mã category |
| 6 | detail | ntext | Mô tả về bài viết |
| 5 | createDate | date | Ngày tạo |
| 6 | modifiedDate | date | Ngày chỉnh sửa |
| 7 | status | bit | Trạng thái |

Bảng “ContentCategory”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã ContentCategory (khóa chính) |
| 2 | name | nvarchar | Tên Mục Content |
| 3 | metatitle | nvarchar | Tên tiêu để chạy SEO |
| 4 | displayOrder | Int | Thứ tự hiển thị |
| 5 | categoryID | int | Mã category |
| 6 | createDate | date | Ngày tạo |
| 7 | modifiedDate | date | Ngày chỉnh sửa |
| 8 | status | bit | Trạng thái |

Bảng “Customer”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã Customer (khóa chính) |
| 2 | email | nvarchar | Email khách hàng |
| 3 | phone | varchar | Số điện thoại khách hàng |
| 4 | address | nvarchar | Địa chỉ khách hàng |
| 5 | Name | nvarchar | Tên khách hàng |
| 6 | createDate | date | Ngày tạo |
| 7 | modifiedDate | date | Ngày chỉnh sửa |
| 8 | status | bit | Trạng thái |

Bảng “AccountAdmin”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã tài khoản Admin |
| 2 | userName | nvarchar | Tên tài khoản |
| 3 | passWord | nvarchar | Mật khẩu |
| 4 | status | bit | Trạng thái kích hoạt |

Bảng “Footer”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã footer |
| 2 | text | nvarchar | Thông tin về footer |
| 3 | link | nvarchar | Link về spa |
| 4 | displayOrder | int | Thứ tự hiển thị |
| 5 | typeId | int | Mã về loại Footer |
| 6 | status | bit | Trạng thái |
| 7 | Icon | nvarchar | Biểu tượng |

Bảng “FooterCategory”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã Mục Footer |
| 2 | name | nvarchar | Tên vị trí mục Footer |

Bảng “IPAdress”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã địa chỉ IP |
| 2 | IP | nvarchar | Địa chỉ IP khách truy cập trang web |
| 3 | date | datetime | Ngày khách truy cập vào trang web |

Bảng “Menu”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã Menu |
| 2 | text | nvarchar | Tên các loại Menu |
| 3 | link | nvarchar | Link về menu spa |
| 4 | displayOrder | int | Thứ tự hiển thị |
| 5 | typeId | int | Mã về loại Menu |
| 6 | status | bit | Trạng thái Menu |

Bảng “MenuType”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã Mục Menu |
| 2 | name | nvarchar | Tên danh mục Menu |

Bảng “Product”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã sản phẩm (khóa chính) |
| 2 | name | nvarchar | Tên sản phẩm |
| 3 | metaTitle | nvarchar | Tên tiêu đề sản phẩm |
| 4 | description | nvarchar | Mô tả về sản phẩm |
| 5 | image | nvarchar | Hình ảnh sản phầm |
| 6 | moreImages | XML | Những hình ảnh sản phẩm khác |
| 7 | price | decimal | giá |
| 8 | quantity | int | Số lượng sản phẩm |
| 9 | idCategory | int | Mã danh mục sản phẩm |
| 10 | createdDate | date | Ngày tạo |
| 11 | modifiDate | date | Ngày sửa |
| 12 | status | bit | Trạng thái |

Bảng “ProductCategory”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã ProductCategory (khóa chính) |
| 2 | name | nvarchar | Tên mục sản phẩm |
| 3 | metatitle | nvarchar | Tên tiêu đề mục sản phẩm |
| 4 | displayOrder | Int | Thứ tự hiển thị |
| 5 | categoryID | int | Mã category |
| 6 | createDate | date | Ngày tạo |
| 7 | modifiedDate | date | Ngày chỉnh sửa |
| 8 | status | bit | Trạng thái |

Bảng “Room”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã phòng |
| 2 | Name | nvarchar | Tên phòng |
| 3 | Status | bit | Trạng thái |

Bảng “Services”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã dịch vụ (khóa chính) |
| 2 | name | nvarchar | Tên dịch vụ |
| 3 | metaTitle | nvarchar | Tên tiêu đề dịch vụ |
| 4 | description | nvarchar | Mô tả về dịch vụ |
| 5 | image | nvarchar | Hình ảnh dịch vụ |
| 6 | moreImages | XML | Những hình ảnh dịch vụ khác |
| 7 | price | decimal | giá dịch vụ |
| 8 | idCategory | int | Mã danh mục dịch vụ |
| 9 | createdDate | date | Ngày tạo |
| 10 | modifiDate | date | Ngày sửa |
| 11 | status | bit | Trạng thái |

Bảng “ServicesCategory”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã ServicesCategory (khóa chính) |
| 2 | name | nvarchar | Tên mục dịch vụ |
| 3 | metatitle | nvarchar | Tên tiêu đề mục dịch vụ |
| 4 | displayOrder | Int | Thứ tự hiển thị |
| 5 | categoryID | int | Mã category |
| 6 | createDate | date | Ngày tạo |
| 7 | modifiedDate | date | Ngày chỉnh sửa |
| 8 | status | bit | Trạng thái |

Bảng “Slide”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã Slide (khóa chính) |
| 2 | image | nvarchar | Ảnh Slide |
| 3 | displayOrder | int | Thứ tự hiển thị |
| 4 | link | Int | Thứ tự hiển thị |
| 5 | title | nvarchar | Tên slide |
| 6 | createDate | date | Ngày tạo |
| 7 | subtitle | date | Phụ đề |
| 8 | status | bit | Trạng thái |
| 9 | TextButton | nvarchar | Button |

Bảng “UsefullInformation”

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | id | int | Mã thông tin thống kê |
| 2 | Name | nvarchar | Tên nội dung thống kê |
| 3 | Value | int | Giá trị thu được |

## 2.5. Thiết kế giao diện:

### **2.5.1. Danh sách các màn hình:**

- Trang đăng nhập Admin

- Trang chủ

- Trang quản lý Admin

- Trang sản phẩm

- Trang dịch vụ

- Trang Booking

- Trang Blog

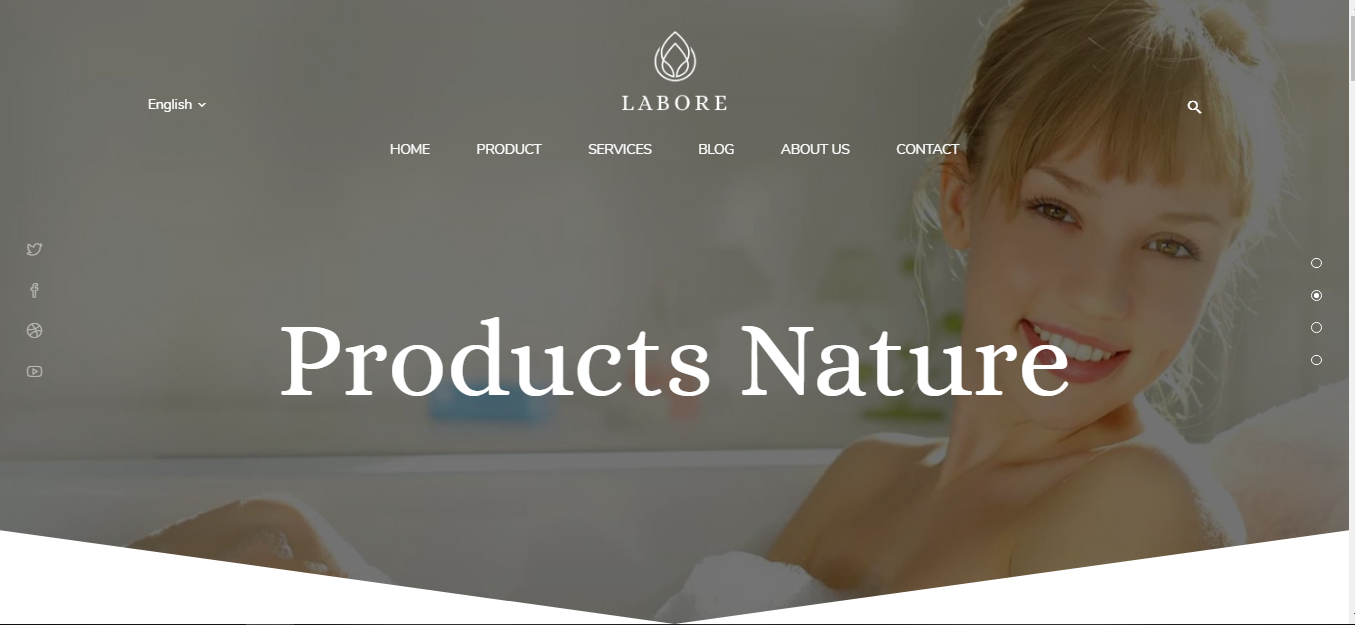
- Trang About Us

- Trang liên hệ Contact

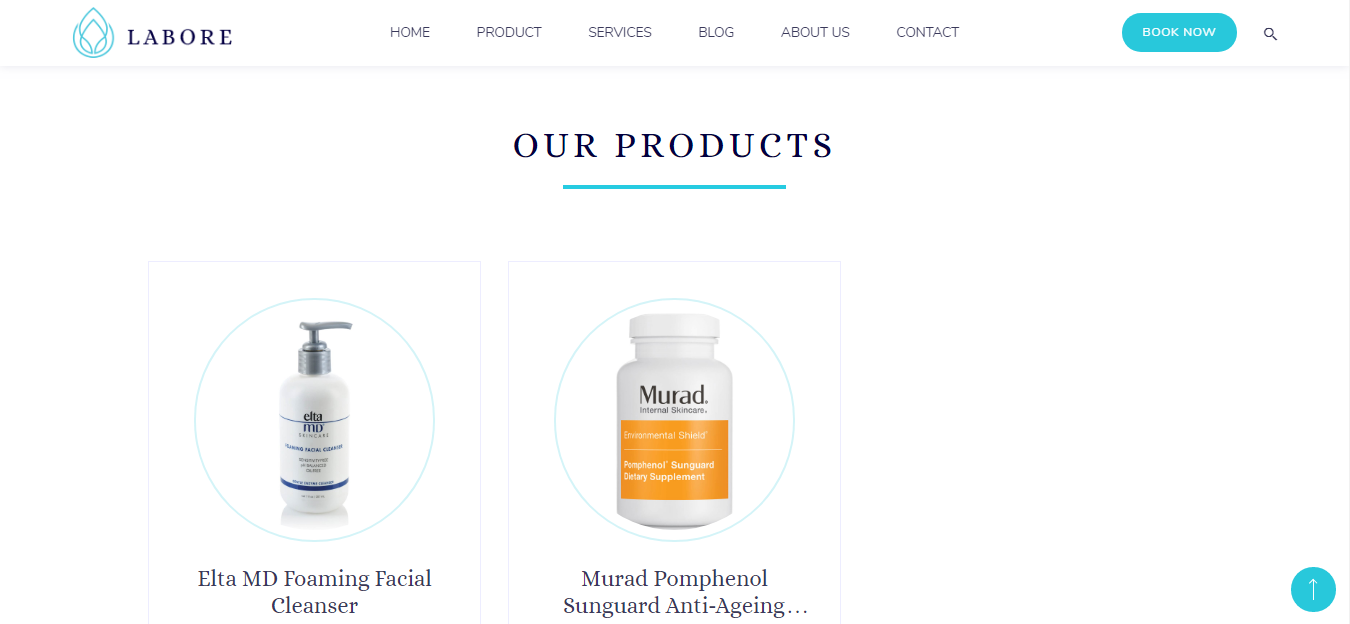
### **2.5.2. Đặc tả các màn hình**

**Nhóm người dùng**

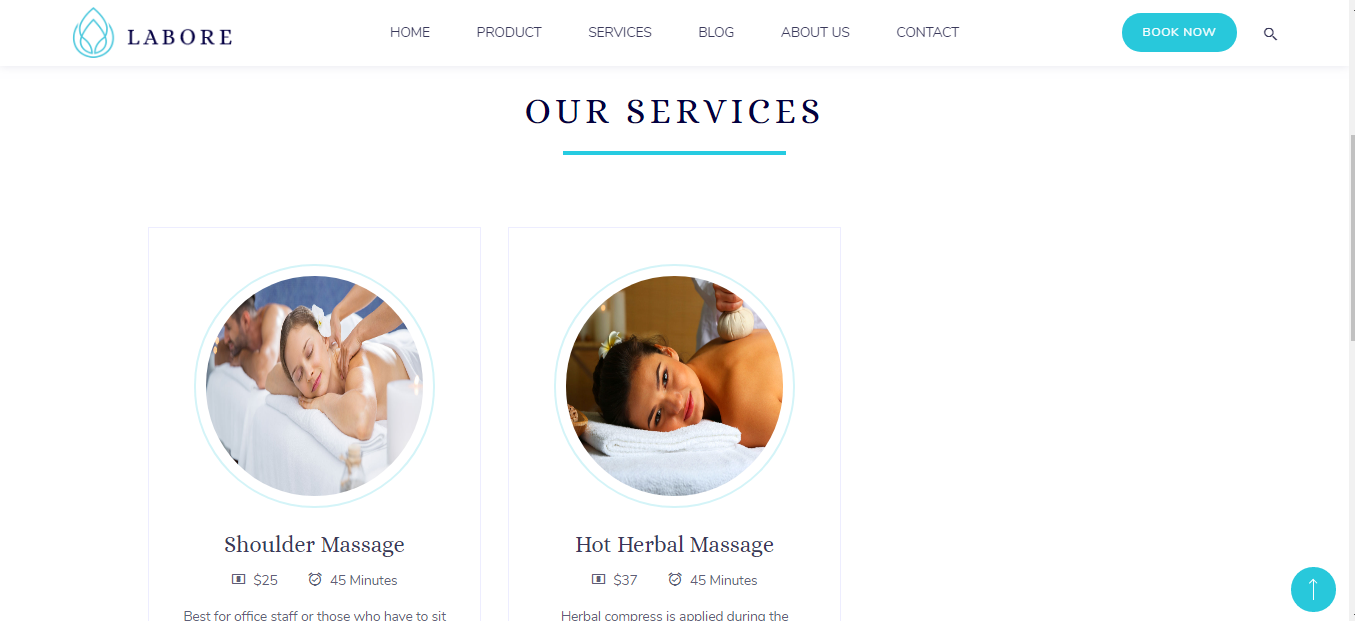
* Trang chủ



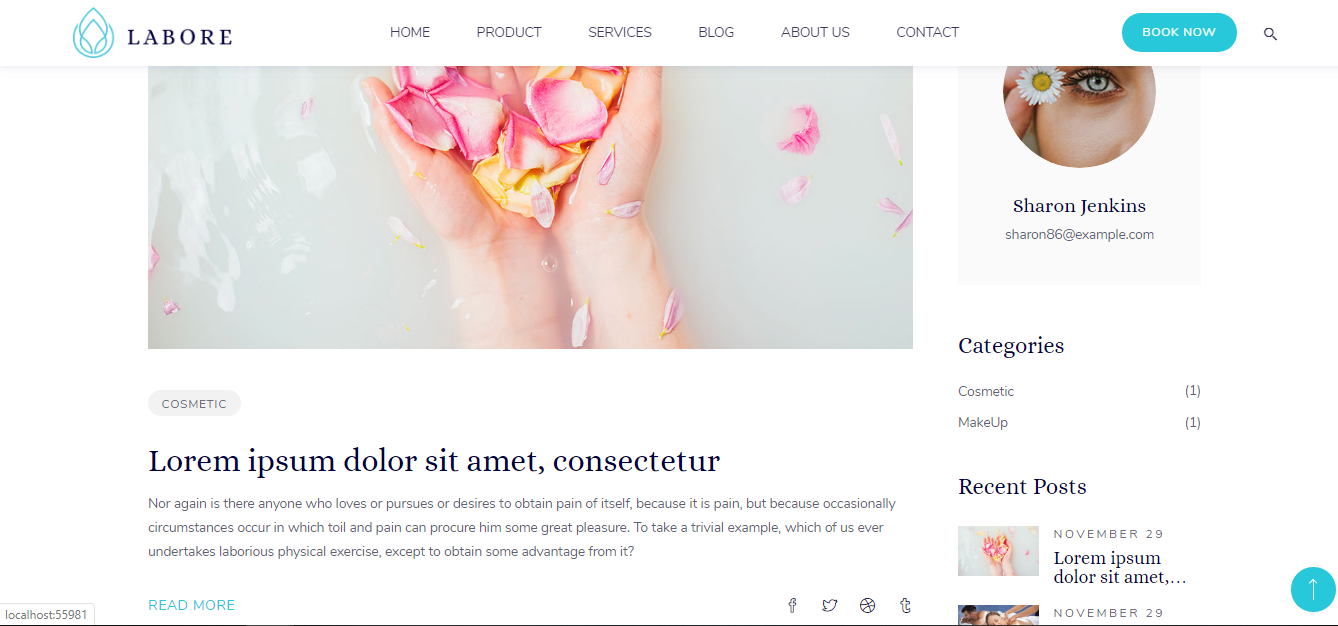
* Trang sản phẩm



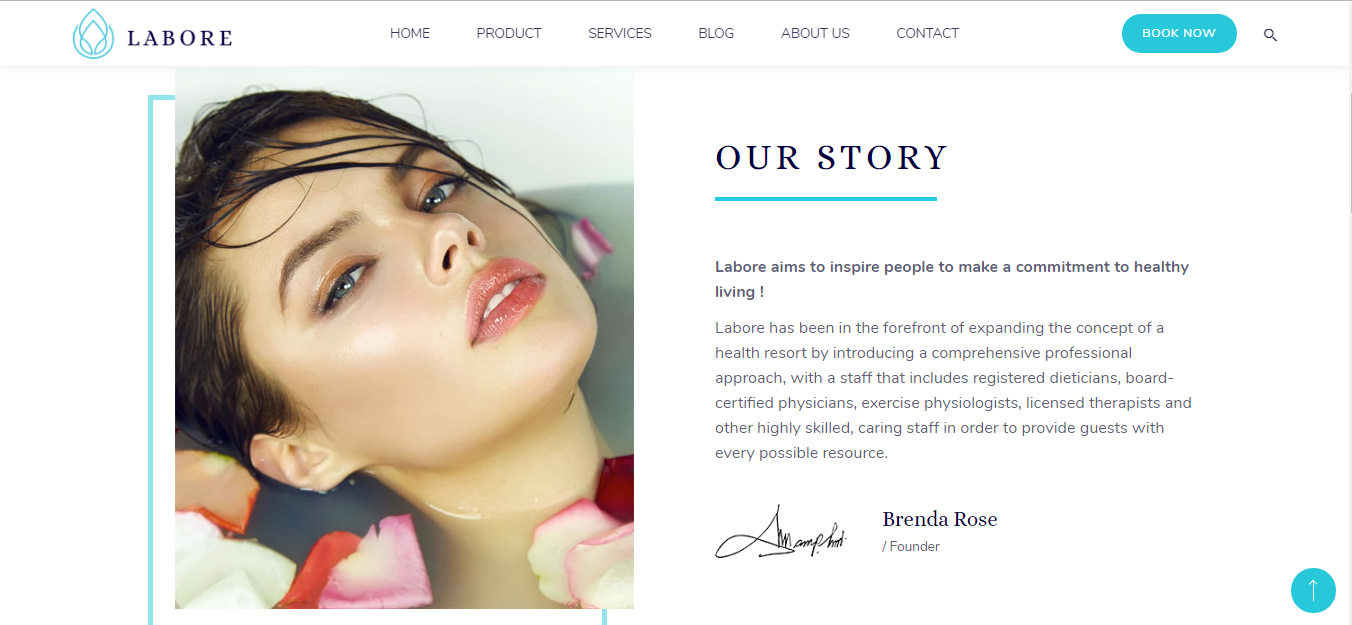
- Trang dịch vụ



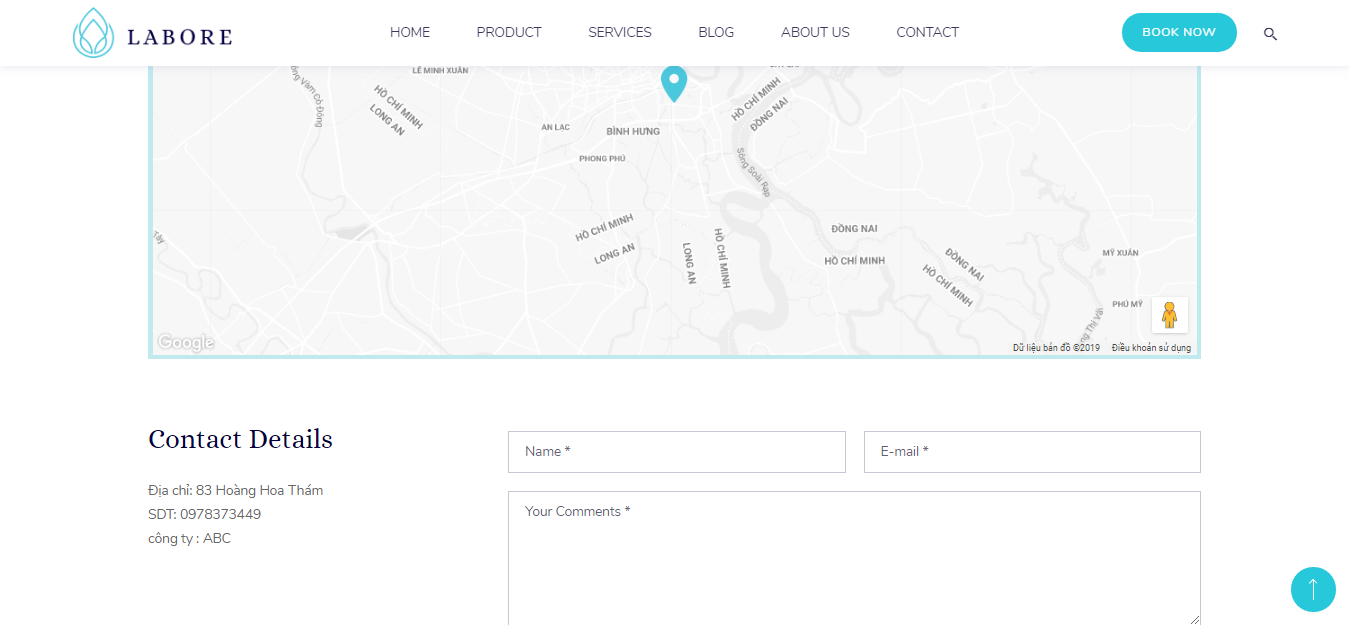
* Trang Blog



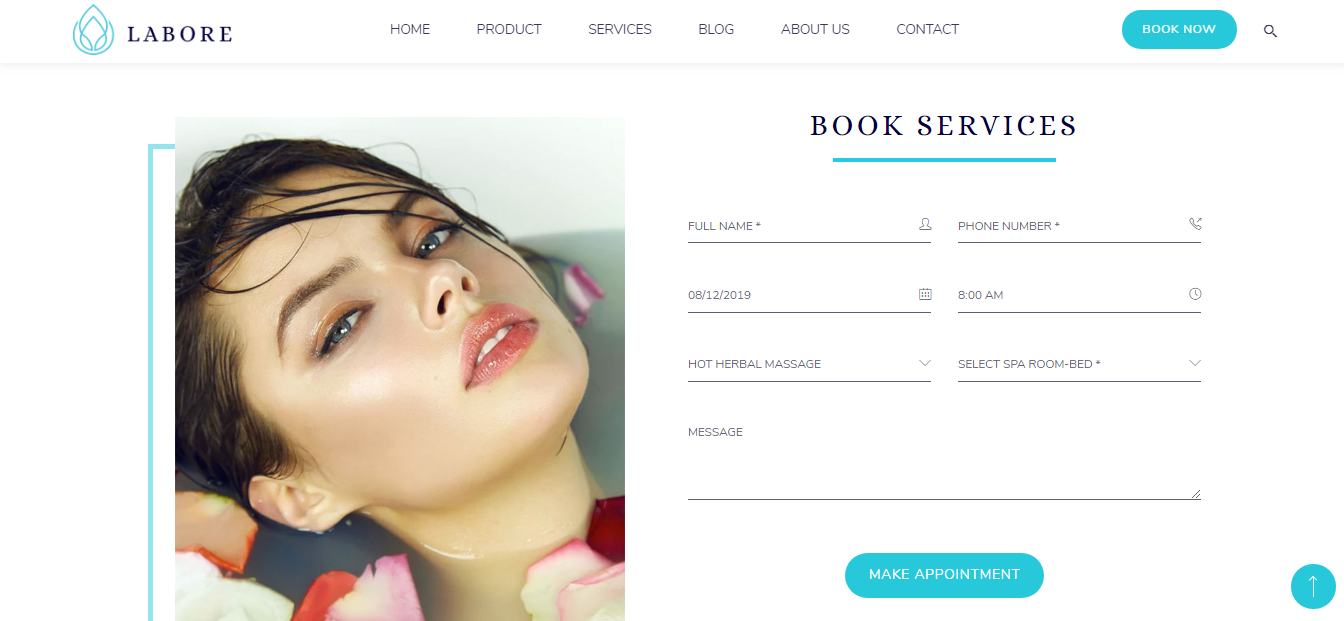
* Trang About Us



* Trang Contact

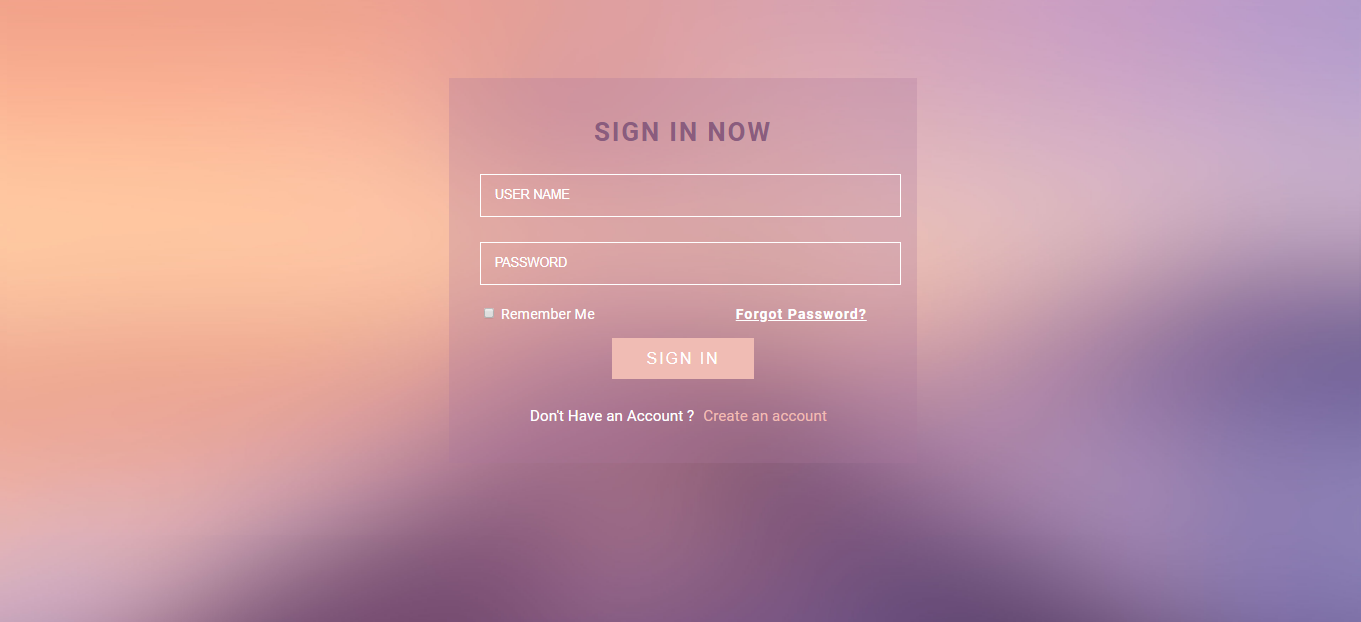


* Trang Booking

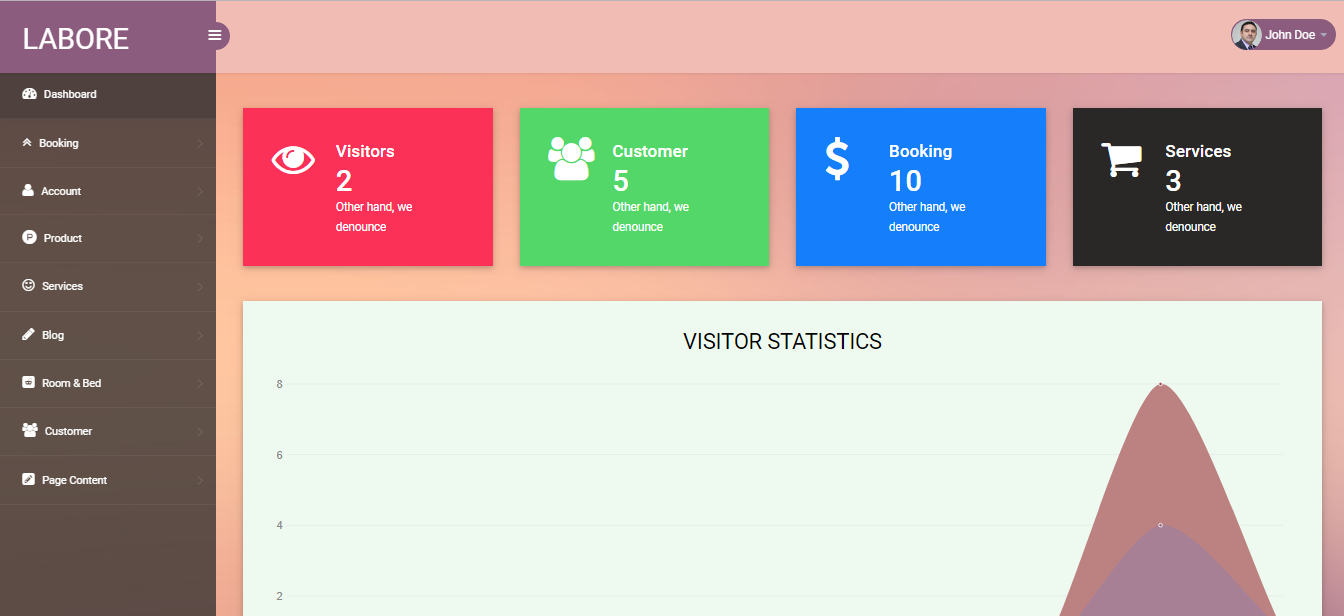


**Nhóm người quản trị**

* Màn hình đăng nhập



* Màn hình trang chủ Admin

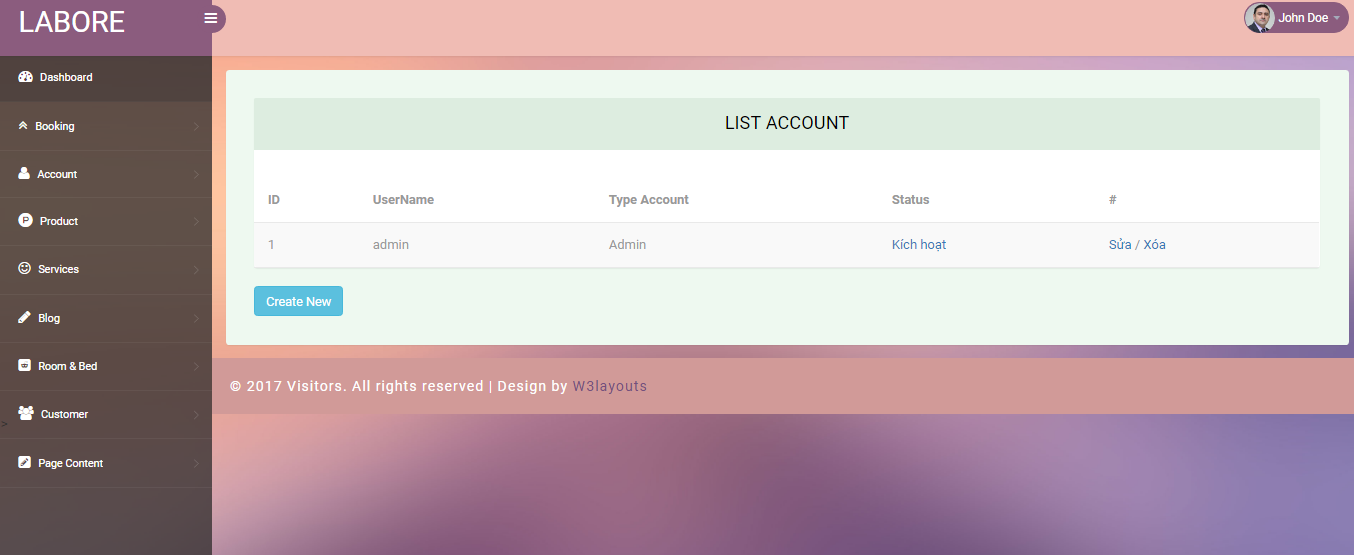


* Một vài màn hình quản lý

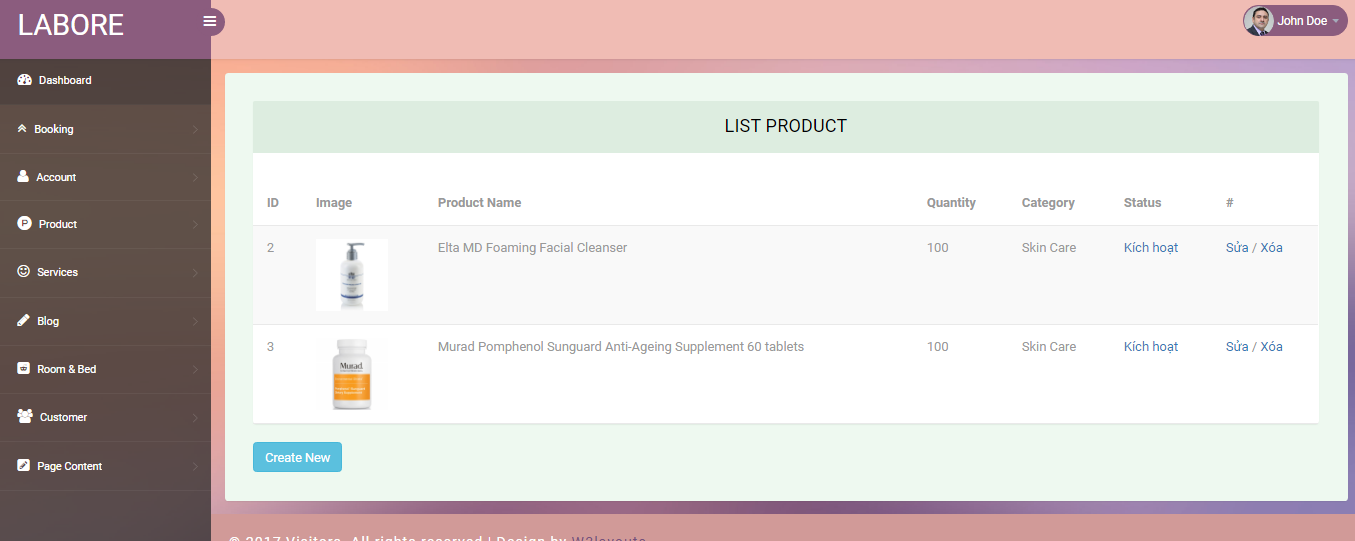
+ Màn hình ListBooking

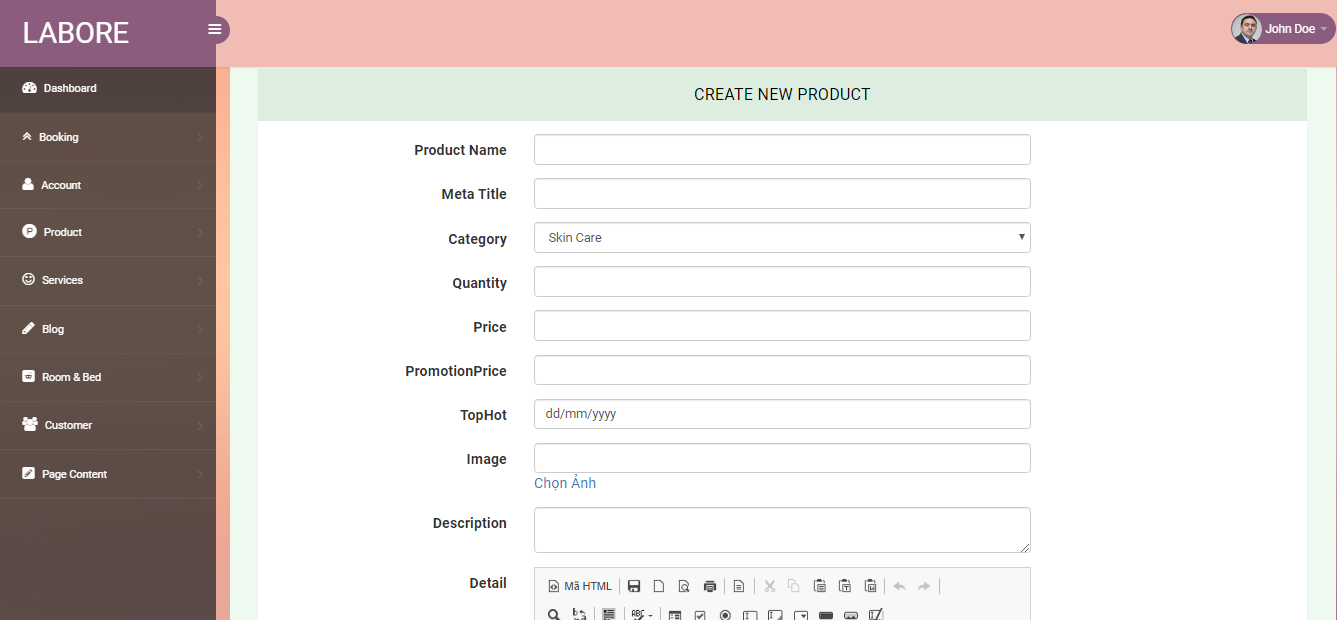


+ Màn hình ListAccount

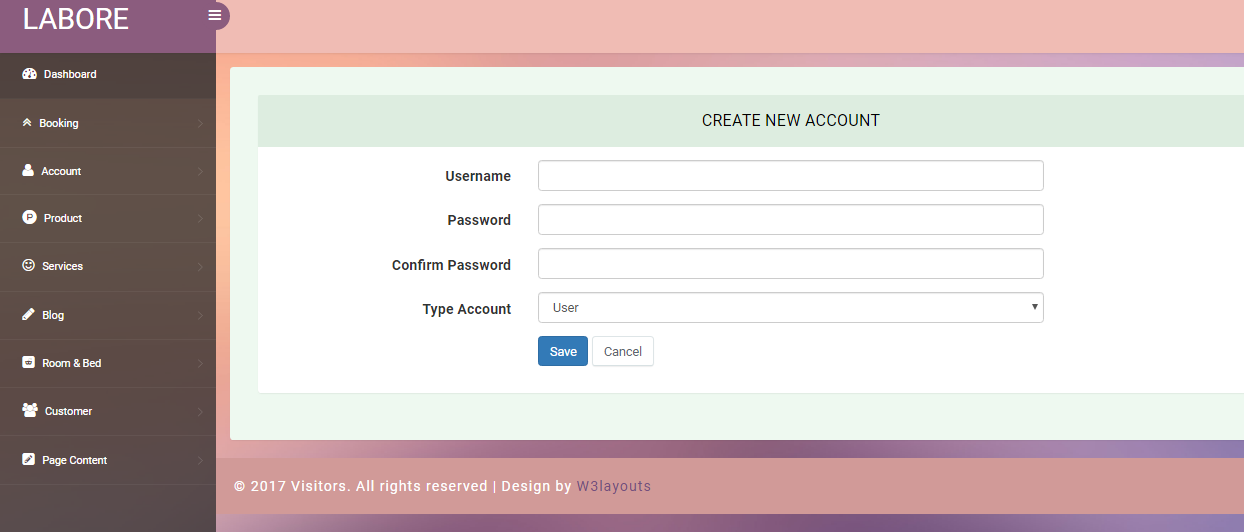


+ Màn hình ListProduct



+ Màn hình Create new product 

+ Màn hình Create New Account



## 2.6. Hiện thực hệ thống

### **2.6.1. Công nghệ sử dụng**

Phía sever:

* Dùng Asp.net MVC 5 để lập trình.
* Cơ sở dữ liệu dùng SQL Sever.

Phía Client: HTML, CSS, Bootstrap và một số thư viện có sẵn.

### **2.6.2. Môi trường lập trình**

Sử dụng công cụ Visual Studio để lập trình quản lý code.

* Visual studio là một trong những công cụ hỗ trợ lập trình và thiết kế website rất nổi tiếng nhất hiện nay của Mcrosoft và chưa có một phần mềm nào có thể thay thế được nó. Visual Studio được viết bằng 2 ngôn ngữ đó chính là C# và VB+.
* Visual Studio cho phép người dùng có thể tự chọn lựa giao diện chính cho máy của mình tùy thuộc vào nhu cầu sử dụng.
* Visual Studio hỗ trợ lập trình trên nhiều ngôn ngữ như: C/C++, C#, F#, Visual Basic, HTML, CSS, JavaScript.
* Là một công cụ hỗ trợ việc Debug một cách dễ dàng và mạnh mẽ như: Break Point, xem giá trị của biến trong quá trình chạy, hỗ trợ debug từng câu lệnh.
* Visual Studio được đông đảo lập trình viên trên thế giới sử dụng.

CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

# **1. Tổng kết**

Trong thời gian bắt đầu nhận đề tài đến nay sau khi trải qua việc phân tích và tìm kiếm hướng giải quyết để thực hiện đề tài thì nhìn chung nhóm đã hoàn thành được các mục tiêu cũng như là những mong muốn mà ban đầu đặt ra. Qua việc tự học và tìm kiếm thông tin qua mạng để thực hiện đề tài, nhóm em đã biết và hiểu hơn về quy trình xây dựng một ứng dụng về cách sử dụng những công cụ hỗ trợ cho việc lập trình, về cách phân tích cơ sở dữ liệu, về việc giải quyết vấn đề… Qua đó giúp nhóm em cải thiện được nhiều kỹ năng hơn và có thêm nhiều kinh nghiệm hơn.

## 1.1. Về mặt lý thuyết

Sau thời gian học tập và thực hiện nhóm đã hiểu về:

* Cách hoạt động và sử dụng thành thạo Asp.net MVC 5
* Cách tạo, sử dụng cơ sở dữ liệu trên hệ quản trị cơ sở dữ liệu MySQL
* Biết cách tích hợp các công nghệ trên để xây dựng thành một ứng dụng quản lý spa chạy trên nền website.

## 1.2. Về mặt hiện thực

Xây dựng website quản lý spa:

* Đối với khách:
* Xem các dịch vụ và sản phẩm chăm sóc spa
* Booking dịch vụ spa
* Đối với người quản trị:
* Chức năng đăng nhập của quản trị
* Chức năng xem các dịch vụ, sản phẩm (có thể thêm, xửa, xóa dịch vụ, sản phẩm
* Chức năng xem các booking của khách hang
* Chức năng chỉnh sửa nội dung trên trang web Client, thêm slide, thêm link social, thêm sản phảm, nội dung, giới thiệu về Spa, Liên hệ.

# **2. Hạn chế**

Nhìn chung website đã đạt được những yêu cầu đề ra ban đầu trong quá trình xây dựng. Sản phẩm website quản lý spa có thể đáp ứng nhu cầu tìm kiếm những dịch vụ và sản phẩm spa của khách hàng. Tuy vậy vẫn còn một vài chức năng cần hoàn thiện để website trở nên thân thiện hơn và đáp ứng một cách tốt nhất với người sử dụng.

# **3. Hướng phát triển đề tài:**

Với lượng thời gian và kiến thức còn hạn chế nên hiện tại đề tài chỉ mang những tính năng cơ bản về spa. Nếu nói về việc đáp ứng hết nhu cầu về spa cho người dùng thì kết quả của đề tài này sẽ không thể bao quát hết được. Do đó hướng phát triển của đề tài này là không thể thiếu. Để có cái nhìn xa hơn trong tương lai và có thể mang lại sự thỏa mãn cho mọi người, đề tài “Xây dựng website quản lý spa” cần phải có các hướng phát triển sau:

* Thêm chức năng tạo tài khoản cho khách hàng.
* Thêm chức năng khách hàng thân thiết
* Thêm chức năng cho khách hang đăn nhập qua mạng xã hội
* Thêm chức năng đề xuất sản phẩm theo xu hướng thị trường hiện tài và thông tin khách hang lấy được qua cache
* Thêm chức năng cho khách hang chọn nhân viên SPA để phục vụ mình
* Thêm chức năng quản lý nhân viên cho ADMIN
* Thêm chức năng Tạo tài khoản cho nhân viên, có thể tạo tài khoản qua mạng xã hội
* Thêm chức năng chia Role dể mỗi tài khoản Admin có thể sử dụng mỗi tính năng hạn chế khác nhau

Với những hướng phát triển trên, nhóm tin rằng trong tương lai ứng dụng khi phát triển ra thị trường người dùng, thì với những tính năng của ứng dụng sẽ giúp cho người dùng có những trải nghiệm tốt hơn trong việc tìm kiếm cho mình một spa phù hợp, tiết kiệm chi phí thời gian tiền bạc cho mọi người, từ đó chất lượng cuộc sống sẽ được nâng cao hơn.

# **4. Tài liệu tham khảo:**

* <https://themeforest.net>[/](https://themeforest.net/)
* <https://w3layouts.com/>