**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙟🕮🙝**

**BÁO CÁO**

**ĐỒ ÁN PHÁT TRIỂN ỨNG DỤNG WEB**



**ĐỀ TÀI:**

**WEB BÁN SÁCH TRỰC TUYẾN**

|  |  |
| --- | --- |
| **Nhóm đề tài** | **: Nhóm 10** |
| **Lớp**  **Giáo viên hướng dẫn** | **: 60.CNTT-1**  **:Bùi Chí Thành** |

**Nha Trang, tháng 3 năm 2021**

**MỤC LỤC**

[**I MỤC TIÊU VÀ NỘI DUNG 6**](#_Toc75127377)

[**1. Mục tiêu 6**](#_Toc75127378)

[**2. Nội dung 6**](#_Toc75127379)

[**2.1 Cơ sở lý thuyết 6**](#_Toc75127380)

[**II CHỨC NĂNG 13**](#_Toc75127381)

1. [**Phân tích yêu cầu 13**](#_Toc75127382)

[**2. Thiết kế cơ sở dữ liệu. 14**](#_Toc75127383)

[**2.1 Mô hinh quan hệ 14**](#_Toc75127384)

[**2.2 Chi tiết các bảng dữ liệu 15**](#_Toc75127385)

[**3.Giao diện trang web. 20**](#_Toc75127386)

[**3.1 Phần User 20**](#_Toc75127387)

[**3.2 Phần Admin 24**](#_Toc75127388)

[**Kết luận phần giao diện Web 25**](#_Toc75127389)

[**III Kết luận. 25**](#_Toc75127390)

[**3.1 Những mặt đạt được 25**](#_Toc75127391)

[**3.2 Những mặt hạn chế 26**](#_Toc75127392)

[**3.3 Hướng phát triển 26**](#_Toc75127393)

[**3.4 Các tài liệu tham khảo: 26**](#_Toc75127394)

**LỜI MỞ ĐẦU**

Thương mại điện tử từ lâu đã không còn là khái niệm mới. Năm 1979 đã chào đón sự ra đời của website mua sắm trực tuyến đầu tiên. Xong so với bây giờ,sau hơn 30 năm phát triển thì sự thay đổi là đáng kể. Những bước chuyển tiến mạnh mẽ về các tính năng như chuyển tiền điện tử, quản lý chuỗi dây chuyền cung ứng,tiếp thị Internet, quá trình giao dịch trực tuyến, trao đổi dữ liệu điện tử, các hệ thống quản lý hàng tồn kho, và các hệ thống tự động thu thập dữ liệu đã được nâng cấp và hợp lý hóa rất nhiều.

Nhận thức rõ tầm quan trọng của thương mại điện tử trong quảng bá thương hiệu hình ảnh, sản phẫm cho doanh nghiệp, đưa đến cho khách hàng cái nhìn chi tiết nhất về doanh nghiệp nên em chọn đề tài ***“Xây dựng website shop bán sách trực tuyến”.***

Website giúp cho khách hàng không mất quá nhiều thời gian, công sức để có thể tìm được một sản phẩm ưng ý. Các thông tin sản phẩm, hình thức thanh toán cũng như giỏ hàng được xây dựng trên hệ thống giúp cho người sử dụng có thể dễ dàng thao tác và thực hiện trực tiếp trên website.

**THÔNG TIN NHÓM 10**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thành viên** | **MSSV** | **Chức vụ** | **Nhiệm vụ** |
| **Nguyễn Văn Hải Long** | **60136035** | **Trưởng nhóm** | **Soạn PowerPoint,vẽ sơ đồ, thiết kế CSDL, Làm Project** |
| **Ngô Minh Thư** | **60137031** | **Thành Viên** | **Thiết kế CSDL, Làm Project** |
| **Lưu Vĩnh Phúc** | **60136556** | **Thành Viên** | **Soạn báo cáo, thiết kế CSDL** |
| **Trần Đinh Trí** | **60131750** | **Thành viên** |  |

1. MỤC TIÊU VÀ NỘI DUNG
2. Mục tiêu

* Nâng cao kiến thức, kỹ năng sử dụng trong Asp.NET MVC và cách thức truy vấn trong SQL Server.
* Ứng dụng xây dựng website giới thiệu và bán sách online với các tính năng như sau:
  + Giao diện thân thiện, dễ sử dụng.
  + Tìm kiếm sản phẩm dễ dàng.
  + Cho phép mua hàng trực tuyến qua mạng.
  + Quản lý sản phẩm đơn hàng trực quan, sinh động.
* Cách tiếp cận : Nghiên cứu Asp.Net MVC
* Sử dụng các phương pháp nghiên cứu: Kiến thức ASP.NET MVC5 hiểu về li html, jQuery và các framework.

1. Nội dung
   1. Cơ sở lý thuyết
      1. Giới thiệu ASP.NET

Đầu năm 2002, Microsoft giới thiệu một kỹ thuật lập trình Web khá mới mẻ với tên gọi ban đầu là ASP+, tên chính thức sau này là ASP.Net. Với ASP.Net, không những không cần đòi hỏi bạn phải biết các tag HTML, thiết kế web, mà nó còn hỗ trợ mạnh lập trình hướng đối tượng trong quá trình xây dựng và phát triển ứng dụng Web.

ASP.Net là kỹ thuật lập trình và phát triển ứng dụng web ở phía Server (Server-side) dựatrên nền tảng của Microsoft .Net Framework.

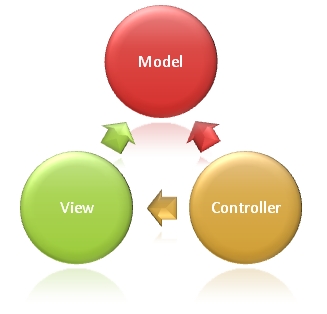
Hầu hết, những người mới đến với lập trình web đều bắt đầu tìm hiểu những kỹ thuật ở phía Client (Client-side) như: HTML, Java Script, CSS (Cascading Style Sheets). Khi Web browser yêu cầu một trang web (trang web sử dụng kỹ thuật client-side), Web server tìm trang web mà Client yêu cầu, sau đó gởi về cho Client. Client nhận kết quả trả về từ Server và hiển thị lên màn hình.

ASP.Net sử dụng kỹ thuật lập trình ở phía server thì hoàn toàn khác, mã lệnh ở phía server (ví dụ: mã lệnh trong trang ASP) sẽ được biên dịch và thi hành tại Web Server. Sau khi được Server đọc, biên dịch và thi hành, kết quả tự động được chuyển sang HTML/JavaScript/CSS và trả về cho Client. Tất cả các xử lý lệnh ASP.Net đều được thực hiện tại Server và do đó, gọi là kỹ thuật lập trình ở phía server.

2.1.2 Giới thiệu ASP.NET MVC Framework

MVC là tên phương pháp chia nhỏ một ứng dụng thành ba thành phần để cài đặt, mỗi thành phần đóng một vai trò khác nhau ảnh hưởng lẫn nhau, đó là model, views, controllers.

MVC là một mẫu thiết kế chuẩn mà nhiều lập trình viên đã quen thuộc. một số loại ứng dụng web sẽ thích hợp với kiến trúc MVC. một số khác vẫn thích hợp với ASP.NET webform và cơ chế postbacks. Đôi khi có những ứng dụng kết hợp với cả hai kiến trúc trên.

Nền tảng MVC bao gồm các thành phần dưới đây :

**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 Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server.

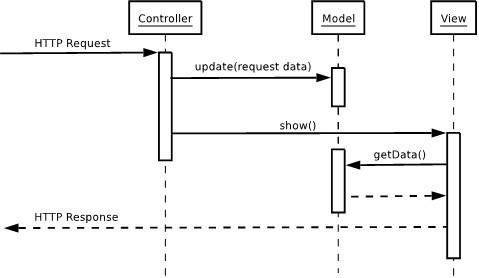
Trong các ứng dụng nhỏ, model thường là chỉ là một khái niệm nhằm phân biệt hơn là được cài đặt thực thụ, ví dụ, nếu ứng dụng chỉ đọc dữ liệu từ CSDL và gởi chúng đến view, ứng dụng khong cần phải có tầng model và các lớp liên quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn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



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.

**Ví dụ:**

Mã:

function getuser($id){

$sql="select \* from info where id='$id'";

$this->query($sql);

$row=$this->fetch();

return $row;}

**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.

2.1.3 Giới thiệu SQL Sever

SQL Server là một hệ thống quản lý cơ sở dữ liệu (Relational Database Management System (RDBMS) ) sử dụng Transact-SQL để trao đổi dữ liệu giữa Client và SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản l. 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 có thể kết hợp với các server khác như Microsoft Internet Information Server (IIS), E -Commerce Server, Proxy Server....

Kết luận phần I

* Qua cơ sở lý thuyết giúp ta nắm được các công nghệ sẽ được sử dụng trong trang web
* Ưu điểm của ASP.NET MVC

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

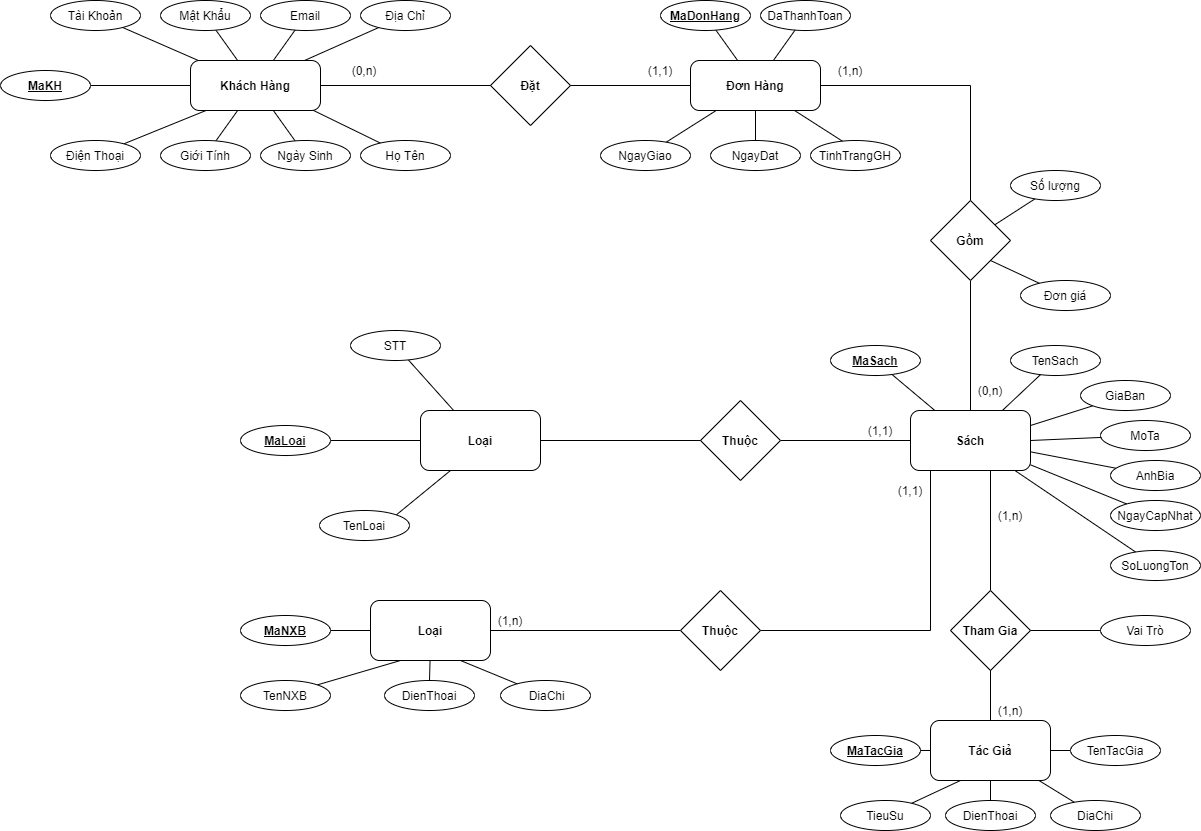
* Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller
* Nó không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
* Nó sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Nhờ đó bạn có thể thiết kế một hạ tầng quản lý định tuyến. Để có nhiều thông tin hơn, bạn nên xem phần [Front Controller](http://go.microsoft.com/fwlink/?LinkId=106357) trên web site MSDN
* Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD)

1. CHỨC NĂNG

1.Phân tích yêu cầu

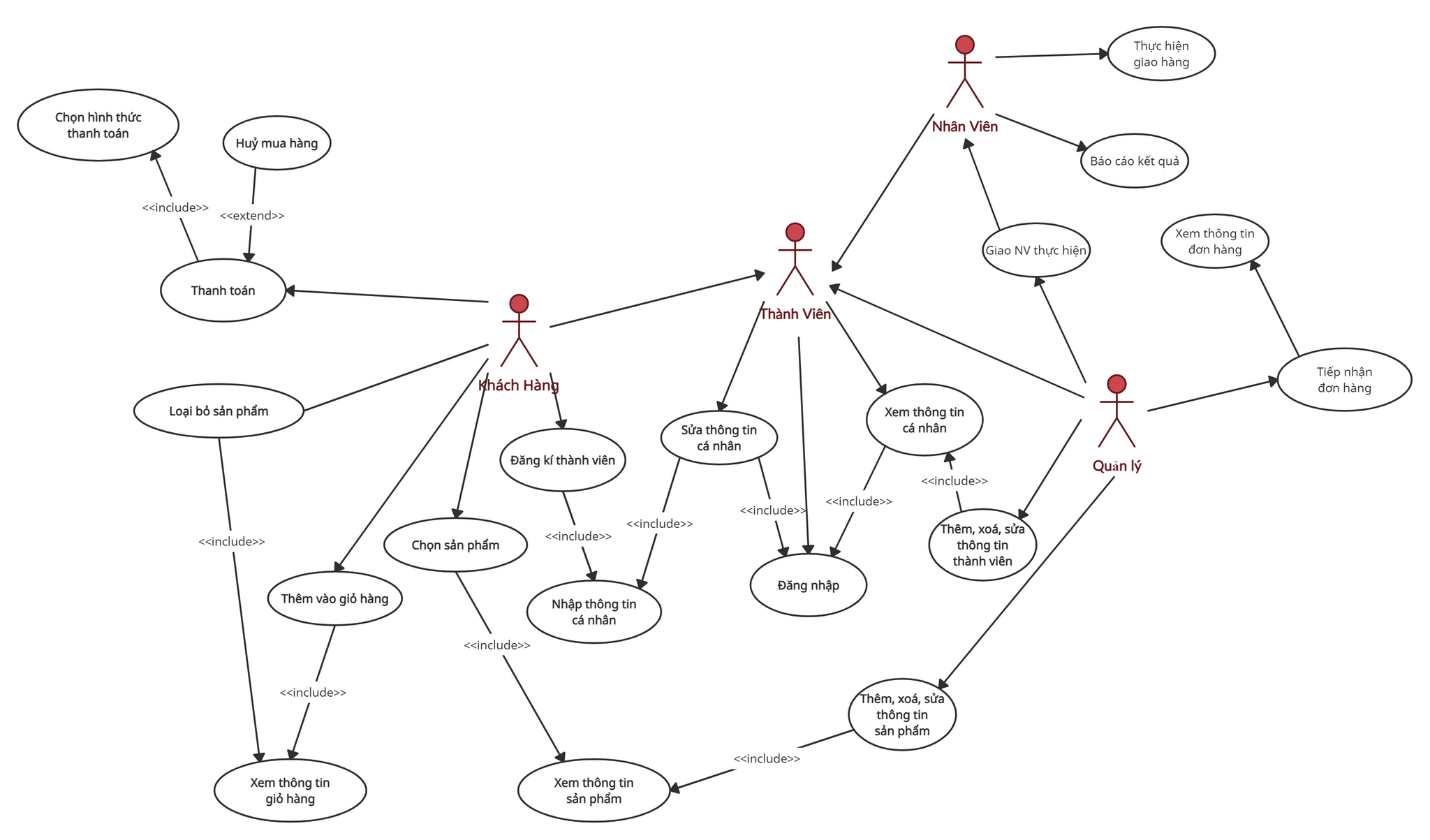
* Phạm vi ứng dụng: Sử dụng trong việc quản lý bán sách của các cửa hàng, siêu thị, doanh nghiệp vừa và nhỏ thực hiện việc nhập và xuất đồ thể thao…
* Đối tượng sử dụng: Phần mềm có thể được sử dụng bởi nhân viên bán hàng, nhân viên nhập hàng, người quản lý, khách hàng mua dụng cụ.

Sơ đồ ER :



*Hình 2.1. Sơ đồ ER*

Sơ đồ Use-case :



*Hình 2.2. Sơ đồ Use-case*

1. Thiết kế cơ sở dữ liệu.
   1. Mô hình quan hệ



### 2.2 chi tiết các bảng dữ liệu

KhachHang(MaKH,HoTen,TaiKhoan,MatKhau,Email,DiachiKH,DienthoaiKH,Ngaysinh)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng** | **Mô tả bảng** | **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** |
| KHACHHANG | Mô tả thông tin khách hàng | MaKH | Mã khách hàng (khóa chính) | Int identity |
| HoTen | Tên khách hàng | Nvarchar(50) |
| TaiKhoan | Tài khoản khách hàng | Varchar(50) |
| MatKhau | Mật khẩu của khách hàng | Varchar(50) |
| Email | Email khách hàng | Varchar(50) |
| DiachiKH | Địa chỉ khách hàng | Nvarchar(200) |
| DienthoaiKH | Số điện thoại của khách hàng | Varchar(50) |
| NgaySinh | Ngày sinh khách hàng | Date |

Loai(MaLoai,TenLoai)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng** | **Mô tả bảng** | **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** |
| Loai | Mô tả loại sách | MaLoai | Mã loại sách | Int |
| TenLoai | Tên loại sách | Nvarchar(50) |

NhaXuatBan(MaNXB,TenNXB,Diachi,DienThoai)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng** | **Mô tả bảng** | **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** |
| NhaXuatBan | Mô tả thông tin nhà xuất bản | MaNXB | Mã nhà xuất bản | Int identity |
| TenNXB | Tên nhà xuất bản | Nvarchar(50) |
| DiaChi | Địa chỉ nhà xuất bản | Varchar(200) |
| DienThoai | Điện thoại của nhà xuất bản | Varchar(50) |

Sach(Masach,Tensach,Giaban,Mota,Anhbia,Ngaycapnhat,Soluongton,MaLoai,MaNXB)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng** | **Mô tả bảng** | **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** |
| SACH | Mô tả thông tin Sách | Masach | Mã sách | Int identity |
| Tensach | Tên sách | Nvarchar(100) |
| Giaban | Giá sách | Decimal(18,0) |
| Mota | Mô tả sách | Nvarchar(max) |
| Anhbia | Ảnh bìa sách | Varchar(50) |
| Ngaycapnhat | Ngày cập nhật | Datetime |
| Soluongton | Số lượng | Int |
|  |  | MaLoai | Mã Loại | Int |
|  |  | MaNXB | Mã nhà xuất bản | int |

Tacgia(MaTG,TenTG,DiaChi,Tieusu,Dienthoai)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng** | **Mô tả bảng** | **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** |
| TACGIA | Mô tả thông tin tác giả | MaTG | Mã tác giả | Int identity |
| TenTG | Tên tác giả | Nvarchar(50) |
| Diachi | Địa chỉ tác giả | Nvarchar(100) |
| Tieusu | Tiểu sử | Nvarchar(max) |
| DienThoai | Điện thoại | Varchar(50) |

Vietsach(MaTG,Masach,Vaitro)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng** | **Mô tả bảng** | **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** |
| VIETSACH | Mô tả thông tin viết sách | MaTG | Mã tác giả | Int |
| Masach | Mã sách | Int |
| Vaitro | Vai trò | Nvarchar |

Dondathang(Madonhang,Dathanhtoan,Tinhtranggiaohang,Ngaydat,Ngaygiao,MaKH)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng** | **Mô tả bảng** | **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** |
| DONDATHANG | Mô tả thông tin đơn đặt hàng | MaDonHang | Mã đơn hàng | Int identity |
| Dathanhtoan | Đã thanh toán | BIT |
| Tinhtranggiaohang | Tình trạng giao hàng | BIT |
| Ngaydat | Ngày đặt | Datetime |
| Ngaygiao | Ngày giao hàng | Datetime |
| MaKH | Mã khách hàng | INT |

Chitietdonhang(MaDonHang,Masach,Soluong,Dongia)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng** | **Mô tả bảng** | **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** |
| CHITIETDONHANG | Mô tả thông tin chi tiết đơn hàng | MaDonHang | Mã đơn hàng | Int |
| Masach | Mã sách | Int |
| Soluong | Số lượng | Int check |
| Dongia | Đơn giá | Decimal(18,0) |

3.2 Chức năng chính của Website

3.2.1 Chức năng đối với khách hàng

a)Xem sản phẩm

|  |  |
| --- | --- |
| Mô tả | Khách hàng có thể xem chi tiết về các loại sách |
| Nhập | Lựa chọn loại sách cần xem |
| Xử lý | Tìm kiếm trong CSDL. |
| Xuất | Hiển thị thông tin về sản phẩm mà khách hàng muốn xem. |

Bảng 3-1: Bảng mô tả chức năng xem sản phẩm

***b) Tìm kiếm***

|  |  |
| --- | --- |
| Mô tả | Khách hàng có thể tìm kiếm sản phẩm hoặc 1 nhóm bằng những form tìm kiếm được xây dựng sẵn trên website. |
| Nhập | Khách hàng nhập vào từ khóa |
| Xử lý | Hệ thống sẽ ghi nhận và thực hiện tìm kiếm trong cơ sở dữ liệu tất cả sản phẩm mà tên có chứa từ khóa hay gần đúng với từ khóa |
| Xuất | Hiển thị thông tin về 1 hoặc nhiều loại sách thỏa mãn được tìm kiếm |

Bảng 3-2: Bảng mô tả chức năng tìm kiếm

***c) Lựa chọn sản phẩm vào giỏ hàng***

|  |  |
| --- | --- |
| Mô tả | Khách hàng có thể lựa chọn loại sách bất kỳ vào giỏ hàng. |
| Nhập | Lựa chọn sách cần mua. |
| Xử lý | -Kiểm tra sản phẩm có trong giỏ hàng.  -Nếu giỏ hàng đã có sản phẩm thì tăng số lượng trong giỏ hàng lên một đơn vị, nếu không thì thêm mới vào giỏ hàng |
| Xuất | Hiển thị thông tin về giỏ hàng mà khách hàng đã lựa chọn. |

Bảng 3-3: Bảng mô tả chức năng chọn sản phẩm vào giỏ hàng

***d) Mua hàng***

|  |  |
| --- | --- |
| Mô tả | Sau khi chọn sản phẩm vào giỏ hàng, khách hàng có thể mua hàng. |
| Nhập | Nhập sản phẩm vào giỏ hàng và cung cấp thông tin họ tên, địa chỉ, email, điện thoại, công ty, fax để đặt hàng. |
| Xử lý | -Kiểm tra sản phẩm có trong giỏ hàng.  -Kiểm tra thông tin khách hàng.  -Ghi nhận hóa đơn khách vào CSDL. |
| Xuất | Hiển thị thông tin về sản phẩm mà khách hàng đã đặt hàng. |

Bảng 3-4: Bảng mô tả chức năng mua hàng

***e) Đăng kí thành viên***

|  |  |
| --- | --- |
| Mô tả | Khách vãng lai có thể đăng ký làm thành viên thông qua một form được xây dựng sẵn. |
| Nhập | Đưa vào các thông tin cá nhân như user, password, địa chỉ email... |
| Xử lý | - Kiểm tra user không được trùng lặp  - Mật khẩu phải gồm ít nhất 6 ký tự.  - Email có đúng định dạng .  ...  - Thêm mới các thông tin khách hàng, mã hóa mật khẩu vào CSDL nếu đúng. Nếu không yêu cầu nhập lại. |
| Xuất | Thông báo tới khách hàng việc đăng ký đã thành công hoặc trở về trang đăng ký nếu có lỗi. |

Bảng 3-6: Bảng mô tả chức năng đăng kí thành viên

***f) Đăng nhập website***

|  |  |
| --- | --- |
| Mô tả | Đăng nhập vào website bằng cách gõ user và password đã đăng ký. |
| Nhập | Cung cấp tên đăng nhập và mật khẩu |
| Xử lý | Truy cập kiểm tra tính chính xác trong CSDL |
| Xuất | Hiển thị thông báo đăng nhập thành công hoặc lỗi và phải đăng nhập lại. |

Bảng 3-7: Bảng mô tả chức năng đăng nhập website

***g) Đăng xuất website***

|  |  |
| --- | --- |
| Mô tả | Đăng xuất khỏi website |
| Nhập | click vào nút đăng xuất trên website |
| Xử lý | - Hủy toàn bộ session của người dùng hiện tại  - Đưa trở về chức năng của khách vãng lai. |
| Xuất | Đưa về giao diện chính dành cho khách vãng lai |

Bảng 3-8: Bảng mô tả chức năng đăng xuất website

3.2.2 Chức năng của Admin

a)Đăng nhập quản trị

|  |  |
| --- | --- |
| Mô tả | Quản trị viên đăng nhập vào hệ thống bằng cách cung cấp tên đăng nhập và mật khẩu vào form đăng nhập được xây dựng sẵn. |
| Đầu vào | - Tên đăng nhập.  - Mật khẩu. |
| Xử lý | - Kiểm tra thông tin nhập vào có hợp lệ hay không.  - Kết nối đến CSDL để kiểm tra tên đăng nhập và mật khẩu. |
| Đầu ra | - Chuyển đến trang quản trị nếu đăng nhập thành công.  - Quay lại trang đăng nhập nếu đăng nhập thất bại. |

Bảng 3-10: Bảng mô tả chức năng đăng nhập quản trị

***b) Đăng xuất quản trị***

|  |  |
| --- | --- |
| Mô tả | Thoát khỏi tài khoản quản trị viên |
| Đầu vào | Click vào biểu tượng đăng xuất từ giao diện. |
| Xử lý | Tiến hành thoát tài khoản, hủy các biến session liên quan đến chức năng quản trị. |
| Đầu ra | Quay lại trang đăng nhập. |

Bảng 3-11:Bảng mô tả chức năng đăng xuất quản trị

Bảng 3-13:Bảng mô tả chức năng quản lý liên hệ

***c)Quản lý sản phẩm***

|  |  |
| --- | --- |
| Mô tả | Quản trị viên có quyền thêm, sửa hoặc xóa sản phẩm. |
| Nhập | Chọn và sử dụng công cụ quản lý sản phẩm. |
| Xử lý | -Thêm:  Kiểm tra tính hợp lệ của thông tin sản phẩm mới  Kiểm tra có tồn tại hay chưa?  Nếu chưa thêm mới sản phẩm vào CSDL.  -Sửa:  Kiểm tra tính hợp lệ của thông tin sản phẩm mới  Kiểm tra có tồn tại hay chưa?  Nếu chưa cập nhật thông tin sản phẩm vào CSDL.  -Xóa:  Nếu sản phẩm đã thuộc một giỏ hàng hoặc đơn hàng thì không được phép xóa.  Ngược lại cho phép xóa sản phẩm. |
| Xuất | Hiện thông báo thành công hoặc lỗi. |

Bảng 3-14:Bảng mô tả chức năng quản lý sản phẩm

3.3 Phân tích thiết kế hệ thống

3.3.1 Các yêu cầu chức năng chính của hệ thống

***a)Yêu cầu lưu trữ***

* Hệ thống cần lưu trữ các thông tin liên quan đến hàng hóa, thông tin đăng nhập, các đơn đặt hàng.
* Thông tin đăng nhập
* Phần quản trị của website cần đảm bảo tính bảo mật do có thể truy cập tới toàn bộ thông tin về hàng hóa và hoạt động kinh doanh.
* Thông tin hàng hóa
* Cần lưu trữ các thông tin sau đây về sách
* Mã sản phẩm.
* Mã loại.
* Tên sản phẩm.
* Ảnh bài.
* Giá.
* Số lượng.
* Trạng thái.
* Thông tin đơn hàng
* Cần lưu trữ các thông tin sau đây về đơn hàng:
* Mã hóa đơn.
* Trạng thái.
* Ngày tạo.
* Ngày duyệt.
* Tổng giá.
* Tên khách hàng.
* Giới tính.
* Địa chỉ .
* Email.
* .Số điện thoại.
* Địa chỉ giao hàng.
* Thời gian giao hàng.

***b) Yêu cầu tính toán***

* Tính tiền tự động cho các sản phẩm mà khách hàng mua. Mỗi khi khách hàng mua một sản phẩm, sản phẩm này sẽ được lưu vào giỏ hàng để khách hàng dễ quản lý. Hệ thống tự động tính tổng giá trị giỏ hàng dựa trên việc lấy giá bán của từng sản phẩm lưu trong cơ sở dữ liệu và số lượng mỗi sản phẩm mà khách hàng mua, từ đó tính tổng tiền của tất cả các mặt hàng được mua để khách hàng theo dõi. Việc tính toán phải được thực hiện hoàn toàn tự động. Nếu khách hàng tiến hành đặt hàng, các kết quả tính toán này phải được lưu trong đơn đặt hàng.
* Kết quả tính toán: Tổng số tiền khách hàng phải trả cho mỗi lần mua.

***c)Yêu cầu tra cứu***

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

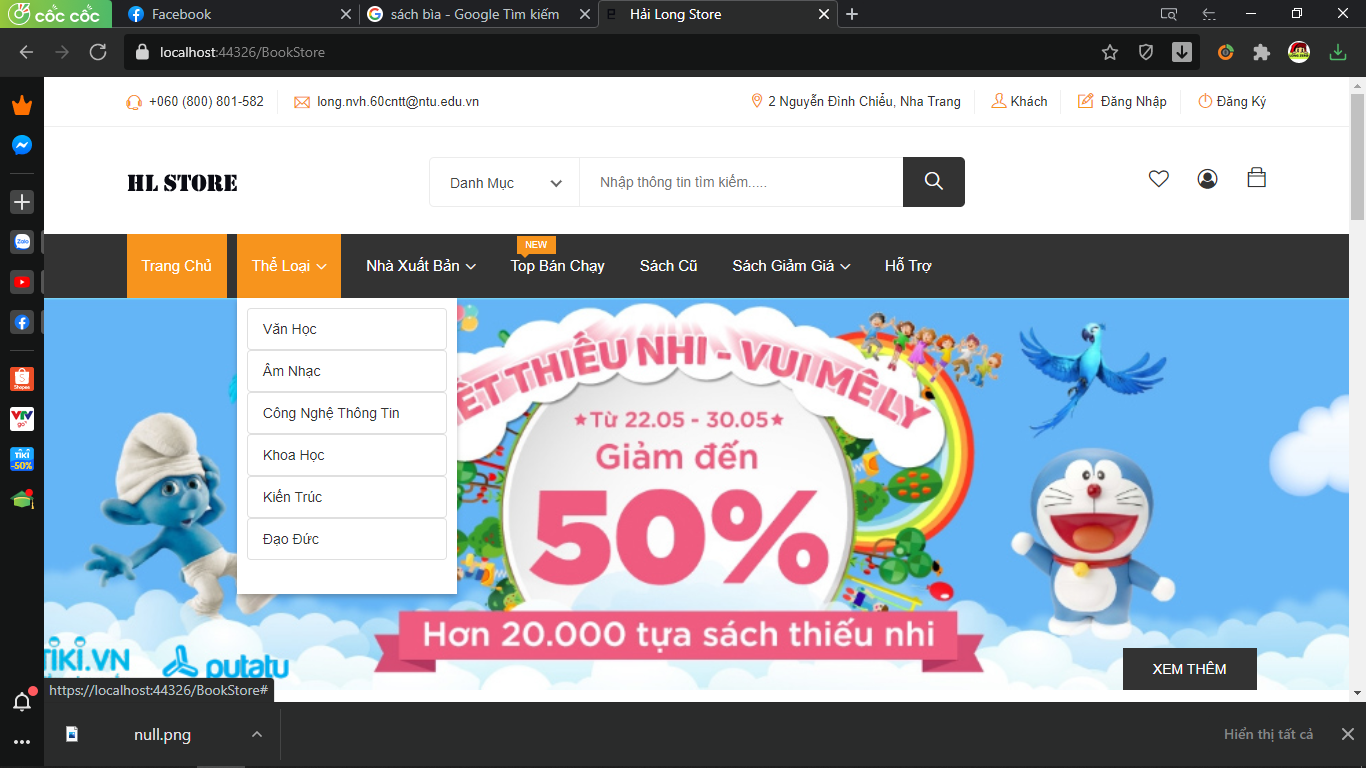
***d)Yêu cầu thống kê, báo cáo***

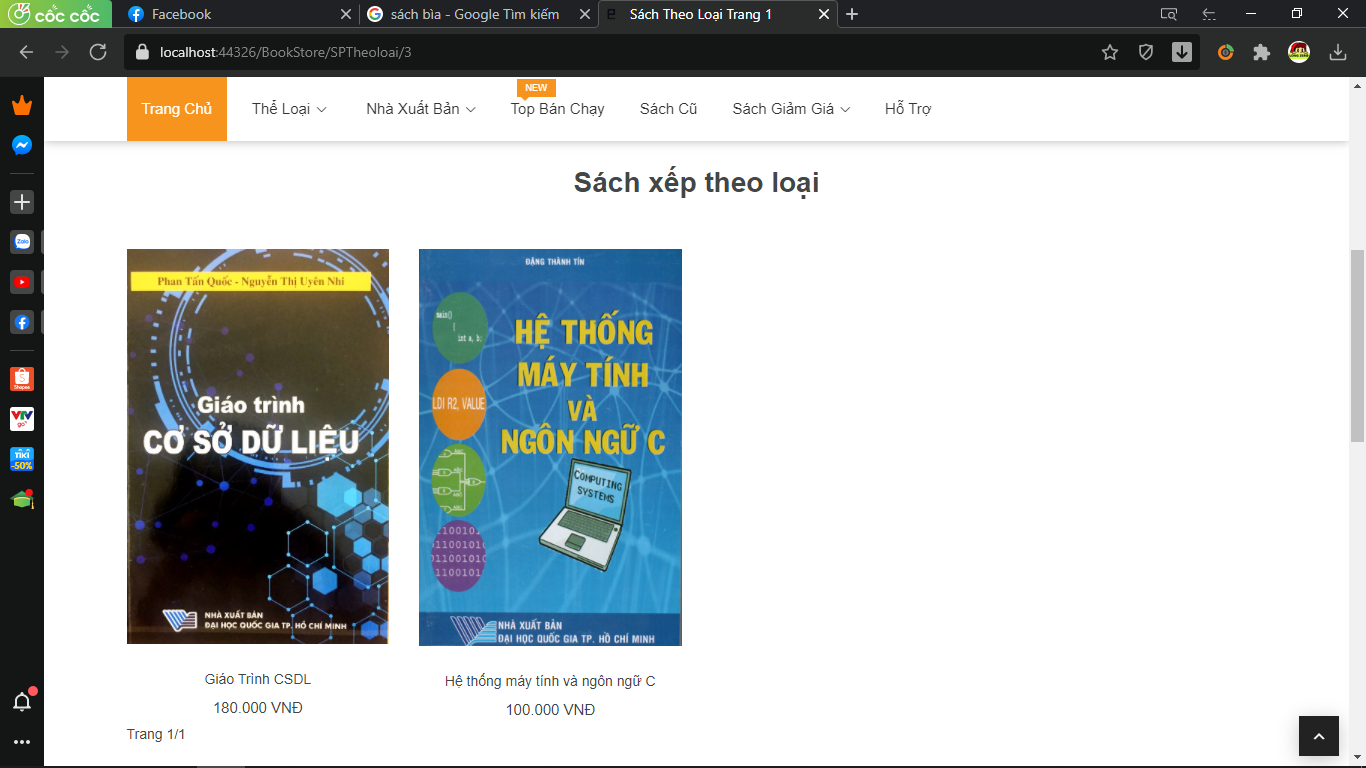
* Thống kê sách bán chạy.
* Thống kê hàng tồn kho.
* Báo cáo doanh thu.

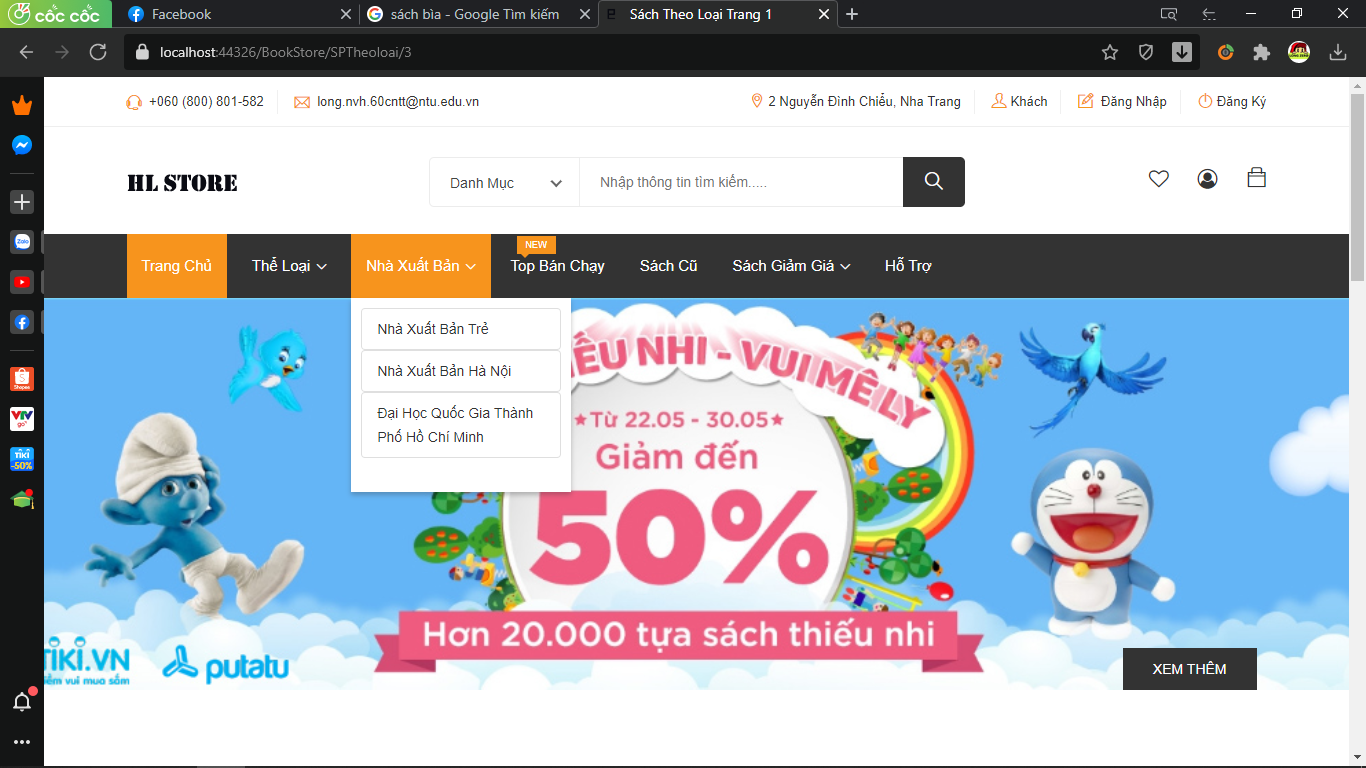
3.Giao diện trang web.

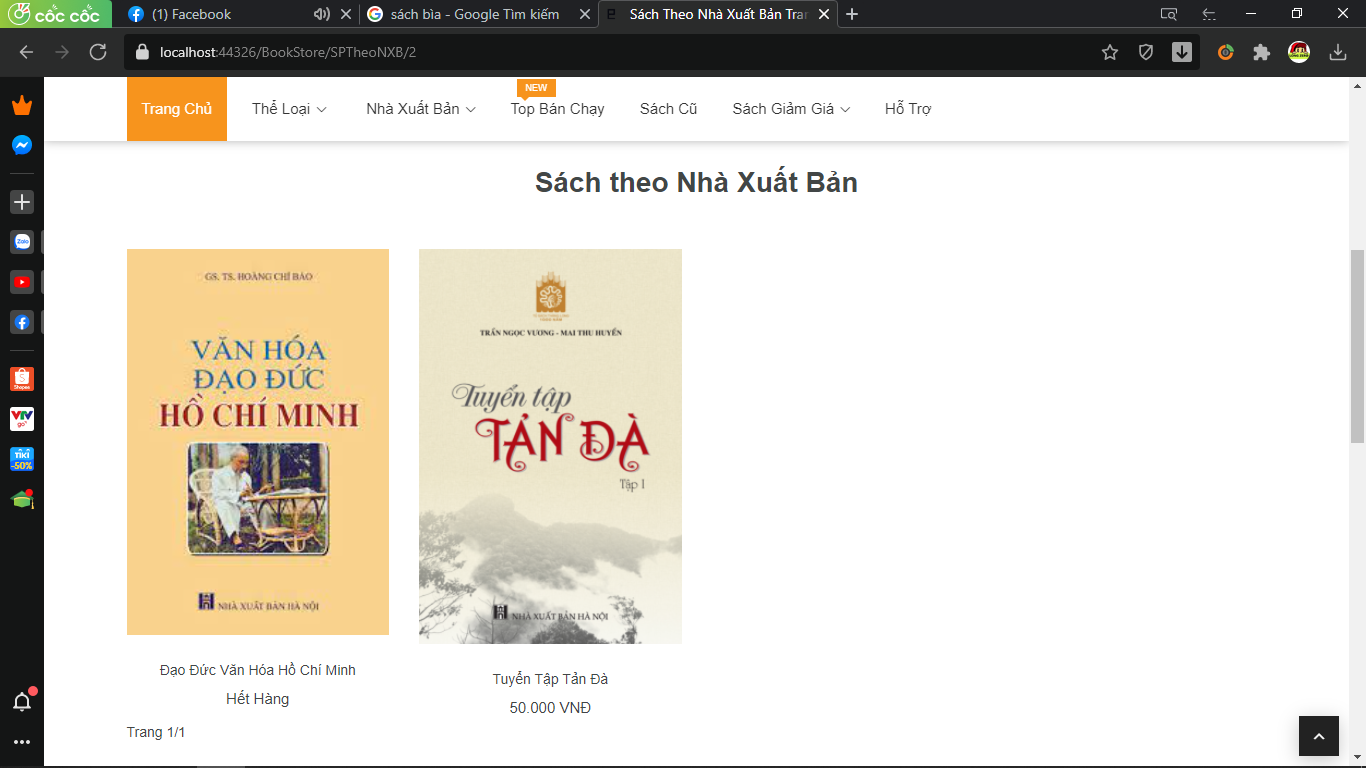
3.1 Phần User

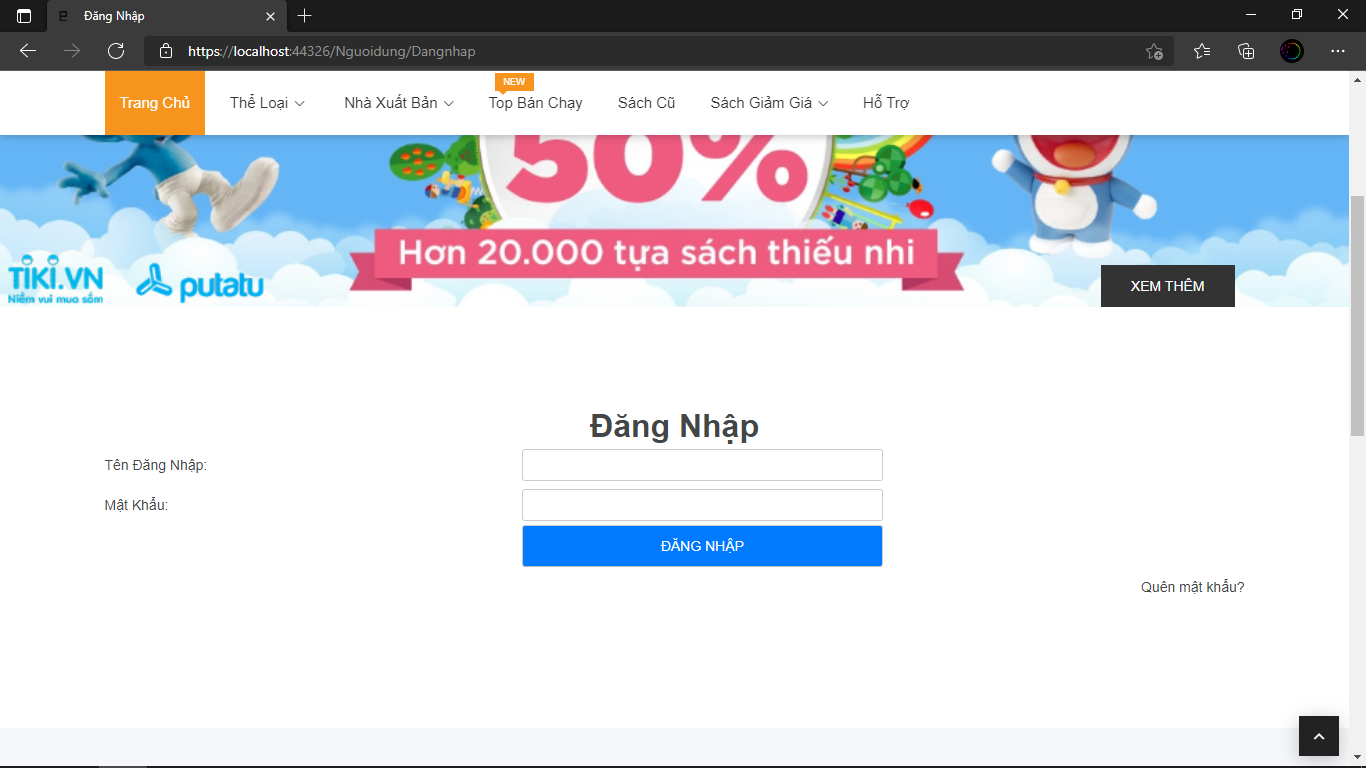
2.1.1 Trang chủ

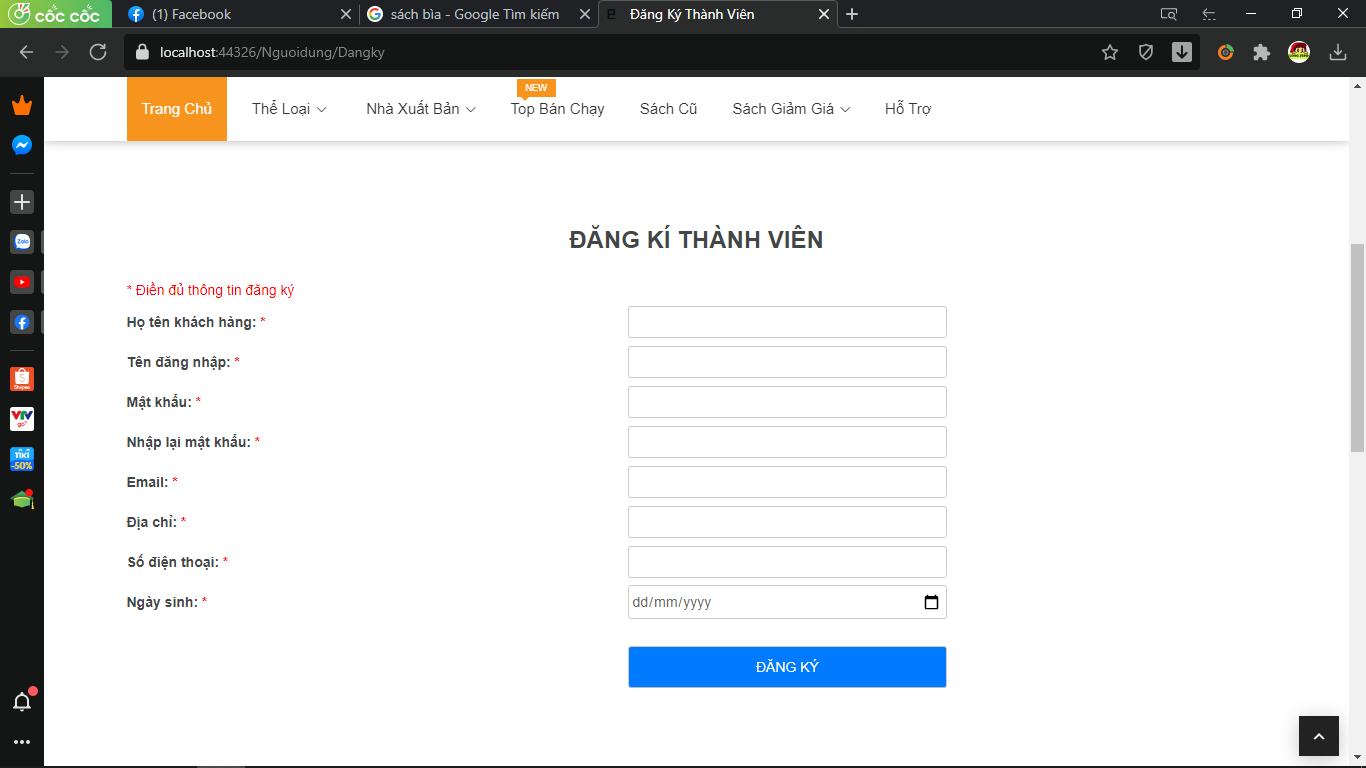


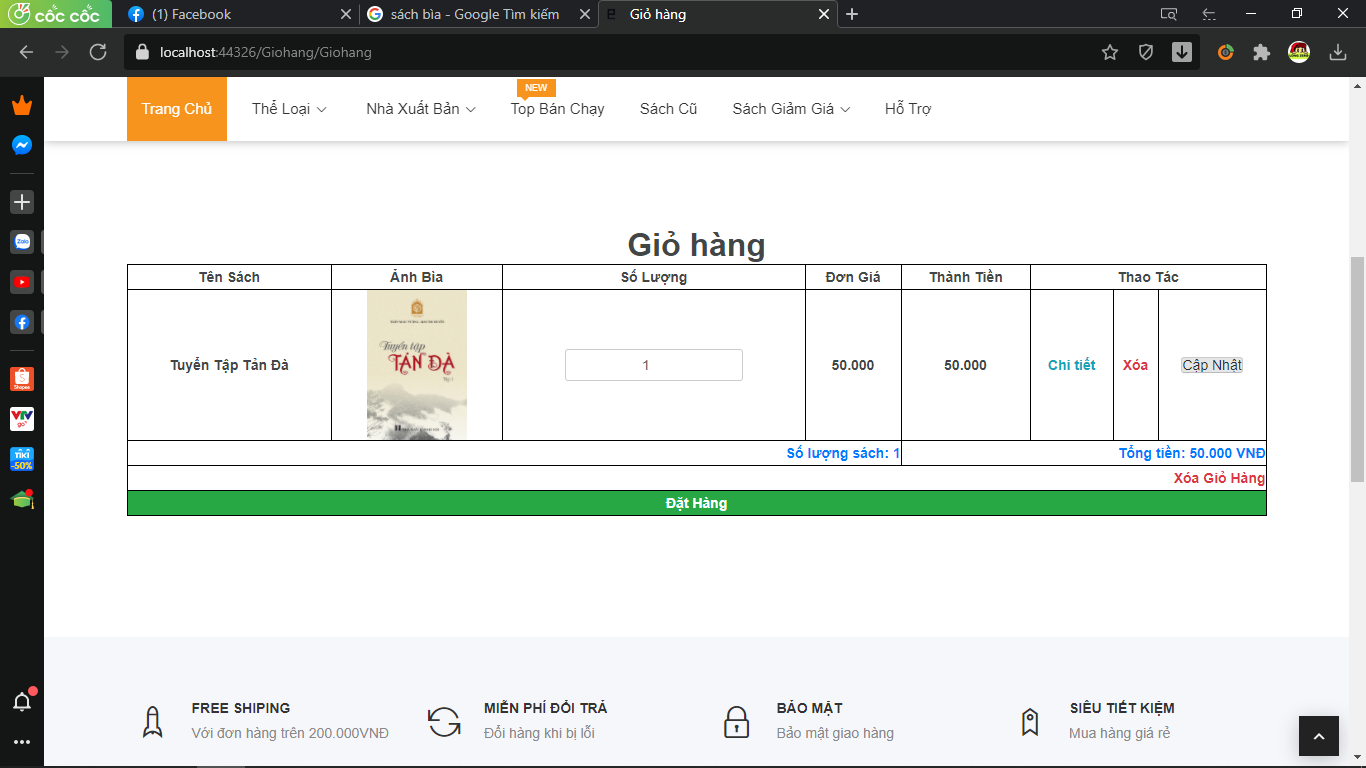
2.2.2 Sách sắp xếp theo loại

2.2.3 Sách sắp xếp theo nhà xuất bản

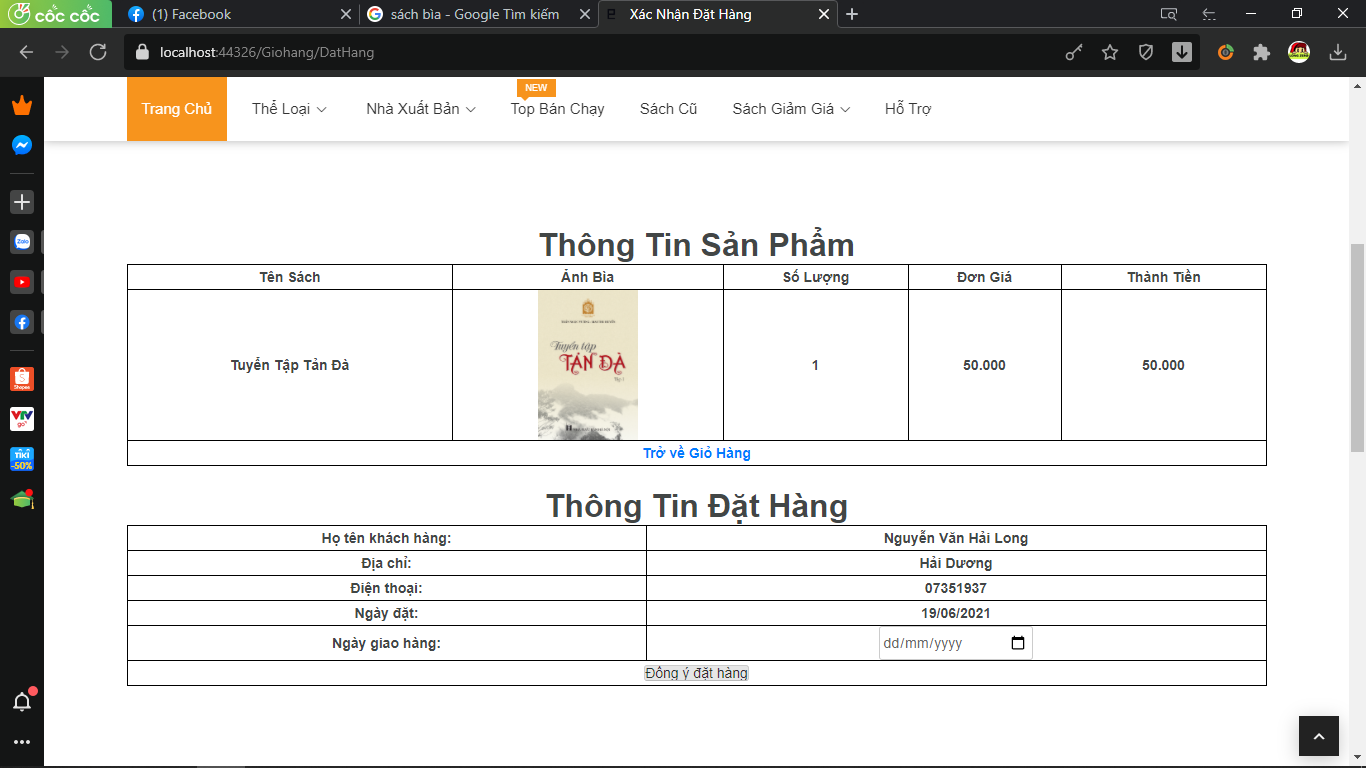


2.2.4 Trang đăng nhập – đăng kí



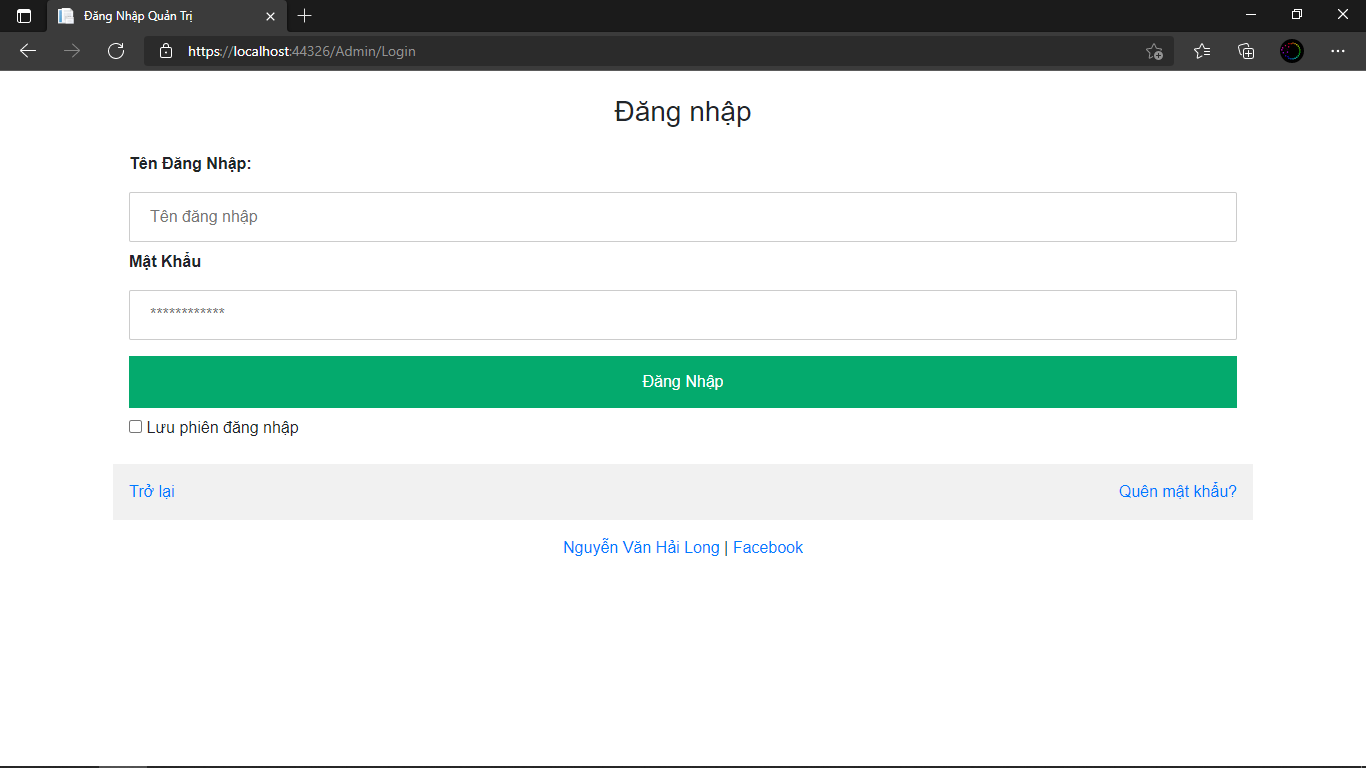
2.2.6 Trang giỏ hàng

2.2.7 Xác nhận đặt hàng

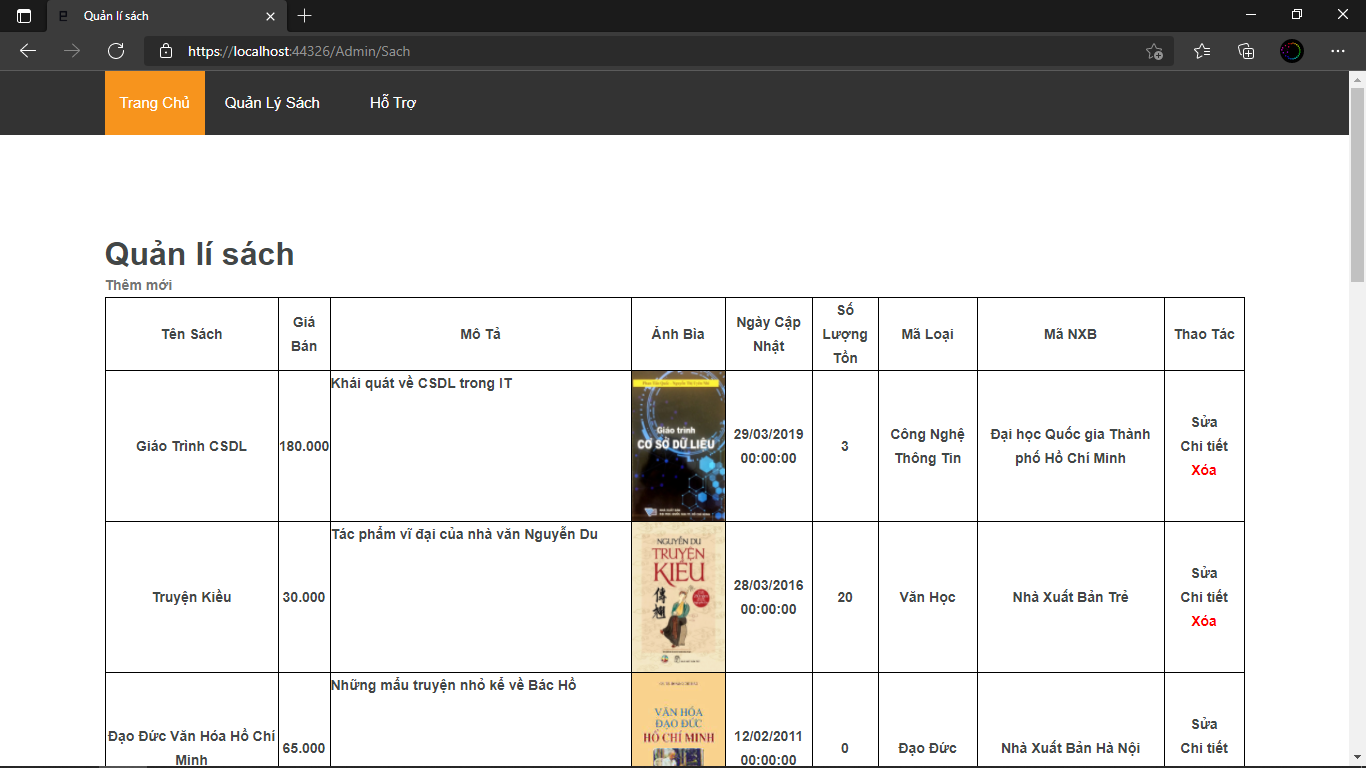


3.2 Phần Admin

3.2.1. Đăng nhập Admin



3.2.2.Thao tác với sách



Kết luận phần giao diện Web

Sau chương này học viên đã thiết kế hoàn tất cơ sở dữ liệu, thiết kế phần giao diện hoàn chỉnh và có thể sử dụng ngay.

1. Kết luận.

3.1 Những mặt đạt được

Thêm xóa sửa trong cơ sở dữ liệu, hiển thị số người đã ghé thăm trang web, số người đang online, hiển thị quảng cáo động, tìm kiếm sách

3.2 Những mặt hạn chế

1. Khi thanh toán chưa có in report.
2. Chủ đề sách và nhà xuất chưa phong phú.

3.3 Hướng phát triển

* Kết nối report khi thanh toán
* Xây dựng hệ thống gửi mail cho khách hàng về những sách mới nhất.
* Kết nối webservice thanh toán trực tuyến qua paypal.

3.4 Các tài liệu tham khảo:

[www.asp.net](http://www.asp.net/)

[www.hmweb.com.vn](http://www.hmweb.com.vn/)