|  |  |
| --- | --- |
| logo (CMYK)-01 | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM**  **KHOA CÔNG NGHỆ THÔNG TIN** |

**BÁO CÁO ĐỒ ÁN JAVA**

**ĐỀ TÀI : WEBSITE KINH DOANH APPLE PRODUCT TRÊN MÔ HÌNH MVC**

TP.Hồ Chí Minh,7/2021

|  |  |  |
| --- | --- | --- |
| GVHD: | Nguyễn Hữu Trung |  |
| SVTH : | Thái Thị Thu Thủy | - 1811063342 |
|  | Nguyễn Thành Long | - 1811063592 |
|  | Hồ Quốc Tuấn | - 1811063447 |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

……………………………………………………………………………………….……………………………………………………………………………………………………………………………………………………………………………….……………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….……………………………………………………………………………………….…………………………………………………………………………………………………………………………

TPHCM, Ngày……Tháng…Năm 2021 Ký tên

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn Thầy Nguyễn Hữu Trung đã tạo điều kiện cho em thực hiện đề tài này.

Chân thành cảm ơn Thầy về những bài giảng trong suốt khóa học, về những kiến thức nền tảng cũng như những kiến thức về chuyên môn đã giúp cho em hoàn thành tốt đề tài.

Trong phạm vi khả năng cho phép, chúng em đã rất cố gắng để hoàn thành đề tài một cách tốt nhất. Song, chắc chắn sẽ không tránh khỏi những thiếu sót. Em kính mong nhận được sự cảm thông và những ý kiến đóng góp của Thầy.

*TpHCM, ngày 30 tháng 7 năm 2021*

Sinh viên thực hiện

**Nguyễn Thành Long**

**Thái Thị Thu Thủy**

**Hồ Quốc Tuấn**

**LỜI MỞ ĐẦU**

Bước sang thế kỉ 21, sự bùng nổ về công nghệ thông tin, thế giới số đã giúp ích cho con người trong rất nhiều lĩnh vực kinh tế, xã hội, quốc phòng, an ninh,… Công việc con người ngày càng phức tạp nhằm đáp ứng nhu cầu cuộc sống ngày càng cao của xã hội, đặc biệt là trong lĩnh vực kinh doanh, mua bán. Bài toán đặt ra là phải làm sao phát triển việc buôn bán trên nhiều nền tảng của một công ty, một cửa hàng kinh doanh một cách chặc chẽ và tiết kiệm thời gian, tiền bạc một cách tiện lợi và hiệu quả mọi lúc mọi nơi nhất. Xuất phát từ nhu cầu thực tiển trên,chúng em đã xây dựng một website bán hàng trực tuyến trên nền tảng ASP.NET với công nghệ MVC5.

**BẢNG PHÂN CÔNG CÔNG VIỆC THỰC HIỆN TRONG ĐỒ ÁN**

|  |  |
| --- | --- |
| HỌ VÀ TÊN | CÔNG VIỆC THỰC HIỆN |
| THÁI THỊ THU THỦY | * THIẾT KẾ GIAO DIỆN CÁC TRANG WEB * PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG CỦA WEBSITE * THIẾT KẾ MÔ HÌNH VÀ HOÀN THÀNH DỮ LIỆU CƠ BẢN TRONG CƠ SỞ DỮ LIỆU * TÌM HIỂU VÀ SOẠN BÁO CÁO ĐỒ ÁN * THIẾT KẾ BÁO CÁO POWERPOINT |
| NGUYỄN THÀNH LONG | * THIẾT KẾ CODE WEB THEO MÔ HÌNH MVC * MODEL: * BUS:   + AccountBUS   + ProductBUS * ViewsModel:   + ProductOfCartModel * VIEW: * Error404: * Cart * Category * Customer * CheckOut * Details * Home * Product: * delProduct\_Cart * insertCart * insertCartWithSL * Shared: * \_Fotter * \_Header * \_Layout * \_LoadProductCart * \_LoginAndSignin * Details\_LoadProductOther * Error * Home\_LoadProductNew * Home\_LoadProductNoiBat * Home\_LoadProductSale * CONTROLER:   + AccountController   + Error404Controller   ProductController |
| HỒ QUỐC TUẤN | * THIẾT KẾ CODE WEB THEO MÔ HÌNH MVC * MODELS:   + CustomerManagerBUS   + DiscountManagerBUS   + OrderManagerBUS   + ProductManagerBUS   + StatisticManagerBUS   CONTROLLERS:   * + CustomerManager   + DiscountManager   + OrderManager   + ProductManager   + StatisticManager * VIEWS:   + CustomerManager:   + ListKH   + DiscountManager:   + addDiscount   + editDiscount   + ListDiscount   + OrderManager:   + ListOrder   + CTOrder   + ProductManager:   + addProduct   + editProduct   + ListProduct   + Shared:   + MainLayoutAdmin   + StatisticManager:     - All |

CHƯƠNG I: TỔNG QUAN 13

1.1. GIỚI THIỆU TỔNG QUAN 13

1.2. MỤC ĐÍCH CHỌN ĐỀ TÀI 13

1.3. PHẠM VI NGHIÊN CỨU 13

1.4. PHƯƠNG PHÁP NGHIÊN CỨU 14

1.5. MỤC TIÊU CẦN ĐẠT 14

1.6 ĐẶC TẢ ĐỀ TÀI 15

CHƯƠNG II: CƠ SỞ LÝ THUYẾT 18

2.1. CÔNG NGHỆ ASP.NET MVC5 18

2.1.1. Lịch sử ra đời: 18

2.1.2 Kiến trúc sử dụng ASP.NET MVC 19

2.2. MÔ HÌNH MVC5 22

2.2.1. Các tính năng của mô hình MVC3 và MVC4 22

2.2.1.1. MVC3 22

2.2.1.2. MVC 4 22

2.2.2. Các cải tiến của MVC5 23

2.2.2.1. Bootstrap 23

2.2.2.2. Authentication Filter 24

2.3. GIỚI THIỆU ENTITY FRAMEWORK 26

2.3.1. Tổng quan 26

2.3.2. Tầng Application 27

2.3.3. Tầng Object Services 27

2.3.4. Tầng EntityClient Data Provider 28

2.3.5. Tầng ADO.NET Data Providers 28

2.3.6. Tầng EDM (Entity Data Model) 29

2.3.7. Cách sử dụng trong Entity Framework 30

2.4. TỔNG QUAN VỀ SQL SERVER 31

2.5. TỔNG QUAN VỀ NGÔN NGỮ GIAO DIỆN 31

2.6.1. HTML và CSHTML 31

2.6.2. Razor View Engine 32

2.6.2.1. Giới thiệu 32

2.6.2.2. Cú pháp Razor 32

2.6.2.2. Khối mã lệnh 33

2.6.3. CSS (Cascading Style Sheets) 33

2.6.4. Bootstrap 34

2.6.5. JavaScript, JQuery và Ajax 34

2.6.5.1. JavaScript 34

2.6.5.2. Jquery 35

2.6.5.3. Ajax 36

2.6.6. Phương thức mã hóa dữ liệu MD5 36

CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 37

3.1. XÁC ĐỊNH ACTOR 37

3.1.1. Quản trị viên hệ thống 37

3.1.2. Người dùng hệ thống (khách hàng) 37

3.2. XÁC ĐỊNH USECASE 37

3.2.1. Quản trị viên hệ thống 37

3.2.2. Người dùng hệ thống 37

3.3. MÔ HÌNH TƯƠNG TÁC NGƯỜI DÙNG VÀ HỆ THỐNG 38

3.3.1. Sơ đồ tổng thể 38

3.3.2. Sơ đồ phân rã Actor 38

3.3.2.1 Actor quản trị viên 38

3.3.2.1.1 Cấu hình giao diện 38

3.3.2.1.2 Quản lí hóa đơn và Cấu hình sản phẩm 39

3.3.2.1.3 Quản lí khuyến mãi 39

3.3.2.1.4 Quản lí hóa đơn 40

3.3.2.2 Actor khách hàng 40

3.3.2.2.1 Quản lí hóa đơn 40

3.3.2.1.Actor Quản trị viên 41

3.3.3. Đặc tả các UseCase tổng quát 41

3.4. MÔ HÌNH CƠ SỞ DỮ LIỆU 41

3.5. Tổ chức bảng dữ liệu 41

3.5.1 Tài Khoản (Taikhoan) 41

3.5.2 Khách Hàng (Khachhang) 42

3.5.3 Danh Mục (Danhmuc) 42

3.5.4 Loại Sản Phẩm (LoaiSP) 42

3.5.5 Sản Phẩm (SanPham) 42

3.5.6 Đơn Hàng (Order) 43

3.5.7 Chi Tiết Đơn Hàng (CT\_Order) 43

3.5.8 Đánh Giá Sản Phẩm (Rate) 44

3.5.9 Giỏ Hàng (Cart) 44

3.5.10 Khuyến Mãi (Discount) 44

3.5.11 Chi Tiết Khuyến Mãi(CT\_DC) 45

CHƯƠNG IV: DEMO VÀ TRIỂN KHAI HỆ THỐNG 46

4.1. Giao diện chức năng: 46

4.1.1 Giao diện chính: 46

4.1.2 giao diện đơn sản phẩm: 46

4.1.3 giao diện danh mục sản phẩm: 47

4.1.4 giao diện chức năng: 47

4.1.4.1 giao diện chức năng đăng nhập: 47

4.1.4.2 giao diện chức năng tạo tài khoản: 48

4.1.4.3 giao diện chức năng quên mật khẩu: 49

4.1.4.4 giao diện chức năng cập nhật thông tin tài khoản: 49

4.1.4.5 giao diện chức năng đặt hàng: 50

CHƯƠNG V: KẾT LUẬN 53

5.1. KẾT QUẢ VÀ KINH NGHIỆM 53

5.1.1. Kết quả 53

5.1.2. Kinh nghiệm 53

5.2. HẠN CHẾ 53

5.3. HƯỚNG PHÁT TRIỂN 53

Tài liệu tham khảo 54

**TÓM TẮT ĐỀ TÀI**

Ngày nay, công nghệ Web ứng dụng trên nền .NET Framework của Microsoft ngày càng phát triển mạnh mẽ cùng với các công nghệ khác như: PHP, JSP,… Trong đó công nghệ ASP.NET MVC là một công nghệ mang nhiều tính năng ưu việt. Bên cạnh đó, nhu cầu thực tế công việc bán hàng ngày online càng phát triển nên đòi hỏi về trình độ đáp ứng cho công việc ngày càng nâng cao và đòi hỏi khả năng xử lý thông tin ngày càng phức tạp nhằm mang lại hiệu quả cao trong công việc bán hàng, thúc đẩy sự phát triển của bán hàng, tiết kiệm thời gian, nhân công, chi phí đầu tư và tính tiện dụng.

Vì vậy, chúng em quyết định chọn đề tài xây dựng hệ thống quản lý bán hàng trên nền tảng Framework ASP.NET MVC5 để hỗ trợ việc bán hàng online của các công ty/cửa hàng bán lẻ để giải quyết các vấn đề đã đề ra trên đó.

Các vấn đề mà chúng em đã nghiên cứu được về Framework ASP.NET MVC5, công nghệ AJAX, cấu trúc JSON, Bootstrap, HTML5. Các hướng tiếp cận phát triển của chúng em là tìm kiếm thông tin qua Internet, xử lý thông tin dưới sự giúp đỡ của Giáo viên hướng dẫn.

Trong quá trình nghiên cứu, xây dựng và phát triển đề tài, em đã gặp nhiều khó khăn và phương án để giải quyết vấn đề là: tham khảo, hỏi ý kiến từ các forum, xin ý kiến đóng góp của giảng viên hướng dẫn, trao đổi thông tin với bạn bè…

Và kết quả đã đạt được, em đã xây dựng thành công website bán hàng đáp ứng đầy đủ yêu cầu đã đặt ra. Hệ thống được vận hành dễ dàng mọi lúc mọi nơi trên mọi thiết bị có hỗ trợ truy cập internet, giao diện tương quan tự động co giản có cấu trúc phù hợp với từng kích thước màn hình hiển thị khác nhau của thiết bị sử dụng.

# CHƯƠNG I: TỔNG QUAN

## 1.1. GIỚI THIỆU TỔNG QUAN

Ngày nay, cùng với sự phát triển vượt bậc của công nghệ thông tin, Microsoft (MS) một trong những đơn vị tuyên phong cho ra đời một nền tảng ứng dựng thế hệ mới chính là WEB APPLICATION (WAPP). WAPP được phát triển trên nền tảng .NET Framework và được vận hành bằng ứng dụng truy cập Website thông thường (Browser) một cách đơn giản, tiện dụng trên mọi thiết bị có hỗ trợ truy cập website và quá trình vận hành được thực hiện mọi lúc, mọi nơi. Đây là một công nghệ rất tiên tiến, đủ để đáp ứng cho nhu cầu công việc ngày càng phát triển, khâu quản lý ngày càng được quan tâm và đặc biệt hơn hết tính tiện dụng đa nền tảng, đa thiết bị ngày càng được quan trọng. Công nghệ WAPP có thể thay thế hoàn toàn cho Winform một nền tảng phát triển ứng dụng cho ra đời các phần mềm được chạy trên hệ điều hành Windows mà hầu hết chúng ta đang được học, phát triển và sử dụng trong thời gian qua.

## 1.2. MỤC ĐÍCH CHỌN ĐỀ TÀI

Nhằm đáp ứng nhu cầu thực tế, xây dựng một Website bán hàng hoạt động độc lập được vận hành mọi lúc mọi nơi, trên mọi thiết bị chỉ cần trang bị một trình duyệt có khả năng truy cập Website.

Nhằm đáp ứng nhu cầu phát triển xã hội, nâng cao năng suất, hiệu quả công việc, tiết kiệm thời gian, nhân công và tiết kiệm chi phí đầu tư trang thiết bị máy móc, đáp ứng nhu cầu chuyên nghiệp hóa xử lý hàng hóa phục vụ cho khách hàng.

Nhằm tạo điều kiện thực hiện nghiên cứu, phát triển khả năng bản thân, góp phần phát triển xã hội.

## 1.3. PHẠM VI NGHIÊN CỨU

Cung cấp nền tảng bán hàng cho một cửa hàng, một công ty

Quy trình khách hàng đặt hàng, mua hàng và nhận hàng, căn cứ vào thông tin thực tế để thiết kế hệ thống thích hợp phục vụ tuần tự đúng cho nhu cầu sử dụng.

## 1.4. PHƯƠNG PHÁP NGHIÊN CỨU

* Quan sát thực tế quy trình bán hàng tại các cửa hàng, công ty
* Trực tiếp đặt hàng tại các cửa hàng/công ty sản phẩm theo quy trình của từng cửa hàng/công ty.
* Nghiên cứu các website mua bán hàng sản có và quy trình thực hiện đặt hàng, mua bán sản phẩm.
* Tham khảo các mô hình, hệ thống trực tiếp sẳn có từ các nước phát triển.

## 1.5. MỤC TIÊU CẦN ĐẠT

* Thêm và quản lý sản phẩm trong hệ thống với số lượng sản phẩm còn lại trong kho của từng sản phẩm, lượt mua, bán của từng sản phẩm.
* Thêm và quản lý mã khuyến mãi được tạo cho từng đợt khuyến mãi với các thông tin khuyến mãi: số lần sử dụng, thời gian áp dụng, khuyến mãi giảm phần trăm hay giảm tiền hàng trực tiếp vào hóa đơn.
* Quản lý các lượt nhập hàng với các thông tin giá nhập vào hệ thống theo từng đợt giá.
* Quản lý và cập nhật liên tục thông tin vận chuyển của các hóa đơn hàng hóa khách hàng đặt hàng có vận chuyển
* Chấp nhận thanh toán bằng nhiều hình thức khác nhau: thanh toán khi nhận hàng, thanh toán trực tiếp khi mua hàng và thanh toán qua chuyển khoản.
* Quản lý các đối tác vận chuyển với các thông tin khai thác của từng đối tác, tích hợp được API các đối tác nhằm mục đích tạo được các đơn vận tự động thông qua hệ thống.
* Quản lý tài khoản ngân hàng của khách hàng, nhân viên công ty, và các nhà cung cấp.
* Quản lý thông tin nhà cung cấp các các sản phẩm kinh doanh.

## 1.6 ĐẶC TẢ ĐỀ TÀI

Đề tài “Xây dựng website bán hàng trên nền tảng công nghệ MVC 5” được hình thành trên ý tưởng ứng dụng công nghệ thông tin vào việc cung cấp nền tảng kinh doanh tại một cửa hàng/công ty bán lẻ góp phần tạo điều kiện thuận lợi để tiết kiệm thời gian, chi phí, nhân công nhằm mang lại hiệu quả cao và chất lượng tốt cho công việc.

Kết quả của quá trình nghiên cứu và thực hiện cho ra đời sản phẩm là website bán hàng Apple Product trực tuyến với tên gọi Error404. Website là một nền tảng tổng hợp các thông tin sản phẩm cần thiết cho một cửa hàng/ công ty kinh doanh. Website hỗ trợ quản lý các thông tin cần thiết:

Khách hàng cần phải có tài khoản đăng nhập thành công để cập nhật địa chỉ giao nhận hàng. Nếu chưa có, khách hàng vẫn có thể mua hàng từ website. Đăng ký tài khoản bao gồm email để nhận mã xác nhận tài khoản và mật khẩu tài khoản. Sau khi xác nhận email thành công, khách hàng phải cung cấp thông tin địa chỉ nhận hàng và sdt để hoàn tất thủ tục đăng kí tài khoản.

Sau khi đăng nhập thành công, khách hàng xem qua các sản phẩm trên trang web. Mỗi sản phẩm thuộc một loại sản phẩm, một loại sản phẩm thuộc một danh mục sản phẩm.

(một danh mục sản phẩm có một hoặc nhiều loại sản phẩm, một loại sản phẩm có một hoặc nhiều sản phẩm).

Khi khách hàng muốn mua một sản phẩm, khách hàng thêm sản phẩm vào giỏ hàng cá nhân (một giỏ hàng có thể có nhiều sản phẩm) và bấm thanh toán.

Trong trường hợp khách hàng chưa đăng nhập, vẫn có thể thêm sản phẩm vào giỏ hàng tạm và bấm thanh toán.

Khi có chương trình khuyến mãi,có thể tạo chương trình khuyến mãi giảm giá trên trực tiếp sản phẩm với các hình thức giảm giá như giảm giá % tiền hàng, giảm giá trực tiếp tiền của hàng, thời gian áp dụng và số lần áp dụng đối với từng tài khoản khách hàng thông qua chương trình khuyến mãi và các áp dụng giảm giá bằng cách trực tiếp điền mã vào hóa đơn khi tạo.

Ngoài ra, hệ thống cho phép lọc sản phẩm với từng mức độ phân loại sản phẩm, phân nhóm sản phẩm, với nhiều đơn vị tính,phân loại giá.

Phần thanh toán :

* Với khách hàng đã đăng nhập vào web và muốn thanh toán:

Web chuyển đến page thông tin thanh toán sản phẩm bao gồm các thông tin cá nhân, địa chỉ nhận hàng KH đã cung cấp và thông tin các sản phẩm khách hàng muốn mua, khách hàng lựa chọn hình thức thanh toán (COD hoặc thanh toán không tiền mặt ), kiểm tra các voucher(hoặc discount), xác nhận số tiền thanh toán và thông tin đặt hàng sau đó mới tiến hành xác nhận đặt hàng.

* Với KH chưa đăng nhập vào web và muốn thanh toán:
* thông qua số điện thoại hoặc email được lấy từ khách hàng, hệ thống sẽ phản hồi khách hàng đã có trong hệ thống hoặc chưa tồn tại, nếu khách hàng chưa có trong hệ thống thì thêm mới thì thông báo khách hàng vào hệ thống tạo tài khoản và tiếp tục quá trình tạo hóa đơn hoặc tự điền các mục thông tin cần thiết để hoàn thành đơn hàng. Khách hàng lựa chọn hình thức thanh toán kiểm tra các voucher(hoặc discount), xác nhận số tiền thanh toán và thông tin đặt hàng sau đó mới tiến hành xác nhận đặt hàng.

Sau khi xác nhận đặt hàng thành công, KH nhận được thông tin đơn đặt hàng bao gồm mã đơn hàng và thông tin liên quan qua email ( và sdt ).

Khách hàng cũng có thể truy cập vào trang web, đăng nhập vào tài khoản và kiểm tra đơn đặt hàng trong mục my account.



*Hình 1: Logo của website*

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT

## 2.1. CÔNG NGHỆ ASP.NET MVC5

### 2.1.1. Lịch sử ra đời:

Tất cả bắt đầu vào những năm 70 của thế kỷ 20, tại phòng thí nghiệm Xerox PARC ở Palo Alto. Sự ra đời của giao diện đồ họa (GUI) và lập trình hướng đối tượng (Object Oriented Programming) cho phép lập trình viên làm việc với những thành phần đồ họa như những đối tượng đồ họa có thuộc tính và phương thức riêng của nó. Không dừng lại ở đó, những nhà nghiên cứu ở Xerox PARC còn đi xa hơn khi cho ra đời cái gọi là kiến trúc MVC (viết tắt của Model – View – Controller).

MVC được phát minh tại Xerox Parc vào những năm 70, bởi TrygveReenskaug. MVC lần đầu tiên xuất hiện công khai là trong Smalltalk-80. Các giấy tờ quan trọng đầu tiên được công bố trên MVC là “A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk – 80”, bởi Glenn Krasner và Stephen Pope, xuất bản trong tháng 8 / tháng 9 năm 1988.

Thế hệ tiếp theo của MVC xuất hiện cùng với hệ điều hành NeXT và các phần mềm của nó.

Kiến trúc này ngày càng được phát triển và hoàn thiện nhằm giải quyết các vấn đề phát sinh cũng như các giải pháp cho quá trình phát triển phần mềm.

Vì vậy sau đó, lần lượt các MVC framework ra đời dựa trên mô hình MVC như: CodeIgniter, Zend, ASP.NET MVC …

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Thời gian phát hành** |  |  | **Phiên bản** |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | **2008** |  |  | ASP.NET MVC 1.0 |  |  |
|  |  |  |  |  |  |  |
| **2008** | |  |  | ASP.NET MVC 2.0 (.NET 3.5) | |  |
|  | |  |  |  | |  |
|  | **1/2010** |  |  | ASP.NET MVC 3.0 (.Net 4) |  |  |
|  |  |  |  |  |  |  |
| **15/8/2012** | |  |  | ASP.NET MVC 4.0 (.Net 4.5) | |  |
|  | |  |  |  | |  |
|  | **17/10/2013** |  |  | ASP.NET MVC 5.0 (.Net 5.0) |  |  |
|  |  |  |  |  |  |  |

*Bảng 1. Lịch sử các phiên bản MVC.*

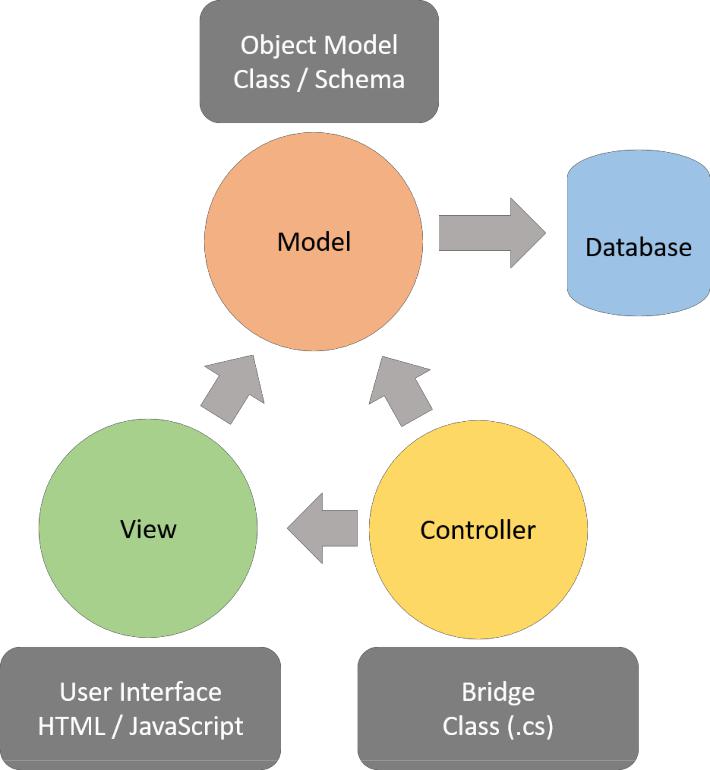
### 2.1.2 Kiến trúc sử dụng ASP.NET MVC

ASP.NET MVC được thiết kế một cách gọn nhẹ, giảm thiểu sự phức tạp của việc xây dựng ứng dụng website bằng cách chia một ứng dụng thành 3 tầng (layer): Model, View và Controller. Sự chia nhỏ này giúp lập trình viên dễ dàng kiểm soát các thành phần trong khi phát triển, cũng như lợi ích lâu dài trong việc kiểm tra, bảo trì và nâng cấp.

**Model**: Model chứa và thể hiện các đặc tính và logic ứng dụng. Theo một cách hiểu khác, Model đại diện cho dữ liệu và logic cốt lõi. Nó chính là những lớp (class) chứa thông tin về các đối tượng mà ta cần phải thao tác, làm việc trên nó. Ví dụ: “Sản phẩm” chứa các thông tin như “Tên sản phẩm”, ”Loại sản phẩm”, “Đơn giá”, …

**View**: View làm nhiệm vụ thể hiện một Model hay nhiều Model một cách trực quan, nó nhận thông tin (một Model hoặc nhiều Model) sau đó biểu diễn lên trang website

**Controller**: Controller nằm giữa tầng View và Model, làm nhiệm vụ tìm kiếm, xử lý một hoặc nhiều Model, sau đó gửi Model tới View để View hiển thị.

*Hình 2. Mô hình MVC*2.1.4 Đặc điểm ASP.NET MVC Framework

Tiếp tục hỗ trợ các tính năng trong ASP.NET

* Hỗ trợ sử dụng các các tập tin:.ASPX, .ASCX, .Master như là thành phần View.
* Hỗ trợ đầy đủ các tính năng bảo mật của ASP.net: Form/ Windows authenticate, URL authorization, membership/roles, output và data caching, section/ profile state, configuration system, provider architecture.

Tách rõ ràng các mối liên quan, mở ra khả năng test TDD (Test Driven Developer).

* Có thể test unit trong ứng dụng mà không cần phải chạy Controllers cùng với tiến trình của ASP.NET và có thể dùng bất kỳ một unit testing framework nào như NUnit, MBUnit, MS Test,…

Có khả năng mở rộng, mọi thứ trong MVC được thiết kế để dễ thay thế, dễ dàng tùy biến. Ánh xạ URL mạnh mẽ, cho phép xây dựng ứng dụng với những URL sạch.

Không sử dụng mô hình post-back từ giao diện gửi đến server. Thay vào đó, chủ động đưa những post-back từ View đến thẳng lớp Controller.

Hỗ trợ nhiều công cụ tạo View (Support for Multiple View Engines)

* Cho phép chọn công cụ tạo view. Hộp thoại New Project cho phép xác định view engine mặc định cho một project.
* Các loại view engine
* Web Forms (ASPX)
* Razor
* Hay một view engine nguồn mở như Spark, NHaml, NDjango.

Hỗ trợ định tuyến

* ASP.NET MVC Framework có một bộ máy ánh xạ URL thật sự mạnh mẽ.
* Bộ máy này cung cấp phương pháp rất linh hoạt trong việc ánh xạ URLs sang các Controller Classes.
* Bạn có thể dễ dàng định ra các quy luật, cài đặt đường đi, ASP.NET dựa vào các quy luật đường đi đó để xác định Controller và action cần phải thực thi.
* ASP.NET còn có khả năng phân tích URL, chuyển các thông số trong URL thành các tham số trong lời gọi hàm của Controller.

Model Binding

* Model Binding là tính năng thế mạnh của ASP.NET MVC (và bây giờ nó cũng được áp dụng cho cả Web Forms trong phiên bản ASP.NET 4.5).
* Hỗ trợ bạn viết phương thức nhận một đối tượng tùy biến như là một tham số
* Với sự hỗ trợ của Model Binding, bây giờ bạn chỉ cần tập trung vào việc cài đặt các nghiệp vụ logic, không cần phải bận tâm về việc suy nghĩ làm cách nào để ánh xạ dữ liệu từ người dùng sang các đối ượng .NET.

Filters

* Là tính năng mạnh trong ASP.NET MVC. Hỗ trợ cho việc kiểm tra tính hợp lệ trước khi một action method được gọi hoặc sau khi một action method thi hành.

Razor View.

Từ ASP.net MVC 3 đi kèm với một công cụ View mới có tên là Razor với những lợi ích sau:

* Cú pháp Razor là sạch sẽ và xúc tích, đòi hỏi một số lượng tối thiểu các tổ hợp phím.
* Việc tìm hiểu Razor tương đối dễ dàng vì nó dựa trên ngôn ngữ C# và Visual Basic.
* Visual Studio bao gồm IntelliSense và mã cú pháp Razor được màu hóa.
* Với Razor views có thể kiểm tra từng đơn vị mà không đòi hỏi bạn phải chạy các ứng dụng hoặc phải chạy website.

## 2.2. MÔ HÌNH MVC5

### 2.2.1. Các tính năng của mô hình MVC3 và MVC4

#### 2.2.1.1. MVC3

So với phiên bản ASP.NET MVC2 thì MVC3 được hỗ trợ thêm HTML5 vàCSS3.

Cải thiện về Model Validation – tính năng kiểm tra tính hợp lệ của dữ liệu do người dùng nhập vào.

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.

#### 2.2.1.2. MVC 4

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 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. Cho phép Logins từ Facebook và những tài khoản khác.
* Phiên bản mới Windows Azure SDK 1.6 được phát hành.

### 2.2.2. Các cải tiến của MVC5

* 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.

#### 2.2.2.1. Bootstrap

Bootstrap là một **framework** CSS được Twitter phát triển. Nó là một tập hợp các bộ chọn, thuộc tính và giá trị có sẵn để giúp web designer tránh việc lặp đi lặp lại trong quá trình tạo ra các class CSS và những đoạn mã HTML giống nhau trong dựán web của mình. Ngoài CSS ra, thì bootstrap còn hỗ trợ các function tiện ích được viết dựa trên JQuery(Carousel, Tooltip, Popovers ,...)

Những lý do để sử dụng Bootstrap:

* Được viết bởi những người có óc thẩm mỹ và tài năng trên khắp thế giới. Sự tương thích của trình duyệt với thiết bị đã được kiểm tra nhiều lần nên có thể tin tưởng kết quả mình làm ra và nhiều khi không cần kiểm tra lại. Vì vậy, giúp cho dự án của bạn tiết kiệm được thời gian và tiền bạc.
* Chỉ cần biết sơ qua HTML, CSS, Javascript, Jquery là bạn có thể sử dụng Bootstrap để tạo nên một trang web sang trọng và đầy đủ. Nhưng lại không cần code quá nhiều CSS.
* Với giao diện mặc định là màu xám bạc sang trọng, hỗ trợ các component thông dụng mà các website hiện nay cần có. Vì nó là opensource nên bạn có thể vào mã nguồn của nó để thay đổi theo ý thích của bản thân.
* Do có sử dụng Grid System nên Bootstrap mặc định hỗ trợ Responsive. Bootstrap được viết theo xu hướng Mobile First tức là ưu tiên giao diện trên Mobile trước. Nên việc sử dụng Bootstrap cho website của bạn sẽ phù hợp với tất cả kých thước màn hình. Nhờ đó mà chúng ta không cần xây dựng thêm một trang web riêng biệt cho mobile.
* Đội ngũ phát triển Bootstrap đã bổ sung thêm tính năng [Customizer.](http://getbootstrap.com/customize/) Giúp cho designer có thể lựa chọn những thuộc tính, component phù hợp với project của họ. Chức năng này giúp ta không cần phải tải toàn bộ mã nguồn về máy.

#### 2.2.2.2. Authentication Filter

Một trong những vấn đề bảo mật cơ bản nhất là đảm bảo những người dùng hợp lệ truy cập vào hệ thống. ASP.NET đưa ra 2 khái niệm: Authentication và Authorize

Authentication là xác thực người dùng, trong MVC5 Authentication Filter được tuỳ chỉnh hoặc chứng thực từ hãng thứ 3 cung cấp.

Khi tạo 1 Project MVC5, người dùng được lựa chọn 4 loại security:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Loại** |  |  | **Mô tả** |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | **No Authentication** |  |  | Ứng dụng không hỗ trợ security |  |  |
|  |  |  |  |  |  |  |
| **Individual User Accounts** | | |  | Ứng dụng sử dụng tài khoản được quản lý | |  |
|  |  |  |  | bởi SQL Server hoặc từ gmail, facebook | |  |
|  | | |  |  | |  |
|  | **Organizational Accounts** |  |  | Ứng dụng sử dụng tài khoản được quản lý |  |  |
|  |  |  |  | bởi Active Directory hoặc Windows |  |  |
|  |  |  |  | Azure Directory |  |  |
|  | | |  |  |  |  |
| **Windown Authentication** | | |  | Ứng dụng chạy trên intranet tức sử dụng | |  |
|  |  |  |  | tài khoản windows để đăng nhập | |  |
|  |  |  |  |  |  |  |

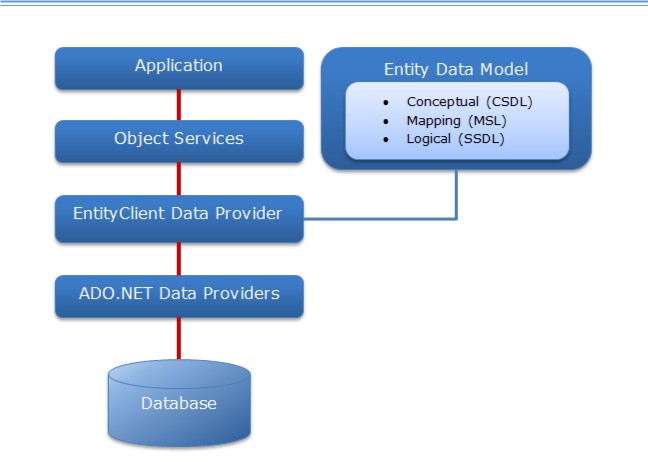
*Bảng 2. Các loại security trong MVC5*

* Mỗi loại security khác nhau sẽ phù hợp với các ứng dụng khác nhau. Khi chọn Individual
* User Accounts thì ứng dụng chạy trên Internet và sử dụng SQL Server để lưu trữ thành viên hoặc đăng nhập từ các hệ thống khác như Google, Facebook... Khi đó người dùng có thể đăng ký, đăng nhập, đổi mật khẩu, đăng xuất với tài khoản cục bộ hoặc từ bên ngoài(Google, Facebook…).
* Code của AccountController có 2 sự khác biệt
* Annotation [Authorize] được sử dụng để làm cho mọi action trong controller không thẻ truy xuất khi chưa đăng nhập ngoại trừ action đó được đánh dấu với annotion [AllowAnontmous]
* Thuộc tính UserManager được tạo ra trong constructor để quản lý các thành viên. Các trang chức năng security đăng ký, đăng nhập và đổi mật khẩu được thực hiện nhờ thuộc tính này.

## 2.3. GIỚI THIỆU ENTITY FRAMEWORK

### 2.3.1. Tổng quan

Entity Framework (EF) là một framework ánh xạ quan hệ đối tượng (ORM) dành cho ADO.NET, là 1 phần của .NET Framework. EF cho phép các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng đặc trưng. Lợi ích lớn nhất của EF là giúp lập trình viên giảm thiểu việc lập trình mã nguồn cần thiết để truy cập và tương tác với cơ sở dữ liệu. EF được Microsoft hỗ trợ phát triển lâu dài và bền vững, vì vậy EF là 1 framework mạnh nhất hiện nay để phát triển ứng dụng Web với sự hỗ trợ đông đảo của các nhà phát triển Web. Kiến trúc của Entity Framework được minh họa như sau:



*Hình 3. Mô hình ki ến trúc Entity Framework*

### 2.3.2. Tầng Application

Application (ứng dụng) là tầng chứa giao diện trang Web (HTML, CSS, Javascript, hình ảnh, …) và các đoạn mã nguồn (C#, VB) để tương tác dữ liệu với các tầng khác trong mô hình thông qua Object Services.

### 2.3.3. Tầng Object Services

Object Services (tạm dịch là các dịch vụ đối tượng) là tầng chứa quá trình tương tác giữa ứng dụng và database, hay nói cách khác nó là nơi chủ yếu để truy cập dữ liệu từ database và trả ngược kết quả về giao diện. Object Services cung cấp các tiện ích để truy vết các thay đổi và quản lý nhận dạng, đồng thời là các quan hệ và thay đổi ở database.

Đây là các class tự động sinh ra tương ứng với mô hình dữ liệu. Các class này bao gồm:

* **ObjectContext** đại diện cho một database. ObjectContext có chức năng quản lý các kết nối, định nghĩa mô hình dữ liệu với metadata và thao tác với database. Lớp nàycũng có thể thêm vào các phương thức đại diện cho các stored procedure trong **ObjectSet<TEntity>** là một một tập hợp các entity. Mỗi đối tượng này tương ứng với một table. Có thể lấy được các đối tượng này thông qua các property tương ứng của ObjectContext.
* **EntityObject, ComplexObject** là các lớp tương ứng cho một dòng dữ liệu của table trong database. Khác biệt chính giữa hai loại này là ComplexObject không chứa primary key.
* **EntityCollection<TEntity> và EntityReference<TEntity>:** là các đối tượng thể hiện mối quan hệ (relationship) giữa hai entity class. Mỗi đối tượng này có thể được truy xuất thông qua các property của entity class.
* Sự ánh xạ tương đương được thể hiện bảng sau:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Database Object** |  |  | **Entity Framework Object** |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | **Database** |  |  | ObjectContext |  |  |
|  |  |  |  |  |  |  |
| **Table, View** | | |  | EntityObject, ComplexObject | |  |
|  | | |  |  | |  |
|  | **Column** |  |  | Property |  |  |
|  |  |  |  |  |  |  |
| **Relationship** | | |  | EntityCollection<TEntity>, | |  |
|  |  |  |  | EntityReference<TEntity> | |  |
|  |  |  |  |  |  |  |

*Bảng 3 Bảng ánh xạ tương đương giữa các đối tượng trong database và EF*

### 2.3.4. Tầng EntityClient Data Provider

EntityClient là một data provider mới của ADO.NET dùng để truy xuất đến database. Được xây dựng bên trên các ADO.NET data provider cơ bản, EntityClient không truy xuất trực tiếp dữ liệu mà thông qua các data provider khác dựa vào các thông tin dữ liệu từ Entity Data Model.

EntityClient cũng bao gồm các lớp giống như các ADO.NET data provider khác và tên lớp được đặt với tiền tố Entity. Ví dụ bạn có thể tạo kết nối bằng EntityConnection, tạo các câu truy vấn bằng EntityCommand và đọc kết quả bằng EntityDataReader.

Một điểm khác biệt với các data provider khác là EntityClient sử dụng Entity SQL để truy vấn dữ liệu. Các lệnh Entity SQL sẽ được chuyển thành một cấu trúc lệnh dạng cây (command tree) và chuyển xuống cho các data provider khác.

### 2.3.5. Tầng ADO.NET Data Providers

Đây là tầng thấp nhấp để dịch các truy vấn L2E (LINQ to Entity) thông qua cây lệnh thành các câu lệnh SQL và thực thi các câu lệnh trong hệ thống DBMS (database management system – hệ quản lý dữ liệu) nào đó. Tầng này kết với database sử dụng ADO.NET.

EntityClient là một data provider mới của ADO.NET dùng để truy xuất đến database. Được xây dựng bên trên các ADO.NET data provider cơ bản, EntityClient không truy xuất trực tiếp dữ liệu mà thông qua các data provider khác dựa vào các thông tin dữ liệu từ Entity Data Model.

EntityClient cũng bao gồm các lớp giống như các ADO.NET data provider khác và tên lớp được đặt với tiền tố Entity. Ví dụ bạn có thể tạo kết nối bằng EntityConnection, tạo các câu truy vấn bằng EntityCommand và đọc kết quả bằng EntityDataReader.

Một điểm khác biệt với các data provider khác là EntityClient sử dụng Entity SQL để truy vấn dữ liệu. Các lệnh Entity SQL sẽ được chuyển thành một cấu trúc lệnh dạng cây (command tree) và chuyển xuống cho các data provider khác.

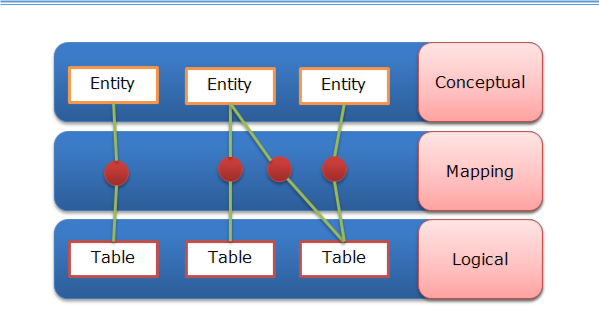
### 2.3.6. Tầng EDM (Entity Data Model)

Entity Data Model (EDM) là mô hình dữ liệu được mô tả thông qua các ngôn ngữ theo chuẩn XML. EDM được chia làm 3 lớp là: Conceptual, Mapping và Logical. Mỗi lớp này được định nghĩa bởi ngôn ngữ riêng theo định dạng XML

Conceptual – Conceptual Schema Definition Language (CSDL): là ngôn ngữ định nghĩa các entity, relationship, hàm trong tập tin với phần mở rộng .csdl. Có thể tạo được các entity class (object layer).

Mapping – Mapping specification language (MSL): định nghĩa các ánh xạ giữa lớp conceptual và logical, nội dung này được lưu trong tập tin .msl.

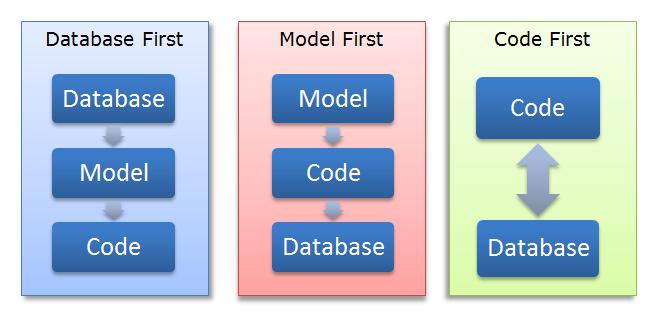
Logical – Store Schema Definition Language (SSDL): định nghĩa mô hình lưu

trữ của dữ liệu, lưu trữ trong tập tin .ssdl.

*Hình 4. Lưu đồ Entity Mapping Model*

### 2.3.7. Cách sử dụng trong Entity Framework

Là thành phần trung gian giữa ứng dụng và database, Entity Data Model (EDM) giúp tạo liên kết và mô tả giữa nguồn dữ liệu vật lý và các đối tượng bussiness (hay object layer) trong ứng dụng. Dựa vào sự lựa chọn thành phần nào sẽ được tạo ra trước, một ứng dụng Entity Framework (EF) có thể sử dụng 1 trong 3 cách tiếp cận: Database First, Model First và Code First.

Các sơ đồ dưới đây minh hoạt mối quan hệ và thứ tự tạo ra giữa các thành phần dữ liệu trong ứng dụng của Entity Framework. Phần Model tương ứng với tập tin .edmx, Code tương ứng với tập tin .cs (hoặc .vb nếu bạn dùng VB.NET).

*Hình 5. Sơ đồ minh họa mối quan hệ các thành phần trong EF*

## 2.4. TỔNG QUAN VỀ SQL SERVER

Là một hệ quản trị cơ sở dữ liệu do Microsoft phát triển. SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ mạng máy tính hoạt động theo mô hình khách chủ cho phép đồng thời cùng lúc có nhiều người dùng truy xuất đến dữ liệu, quản lý việc truy nhập hợp lệ và các quyền hạn của từng người dùng trên mạng.

## 2.5. TỔNG QUAN VỀ NGÔN NGỮ GIAO DIỆN

### 2.6.1. HTML và CSHTML

HTML viết tắt của HyperText Markup Language (tạm dịch Ngôn ngữ Đánh dấu Siêu văn bản) là ngôn ngữ đánh dấu được thiết kế để tạo nên các trang web với các mẫu tạo thành bộ ba nền tảng kỹ thuật cho World Wide Web. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho trang web.

Trong view của ASP.NET, se ̃sử dụng các thẻ HTML đồng thời có thể kết hợp với các khối lệnh ngôn ngữ C# bằng cách khai báo @{…} hoặc @<tên biến> để tạo ra một view template (file .cshtml).

### 2.6.2. Razor View Engine

#### 2.6.2.1. Giới thiệu

Như chúng ta đã biết, từ phiên bản Asp.net MVC 3, Microsoft lựa chọn View Engine mặc định là Razor View Engine, khi View được yêu cầu bởi Http Request thì Razor View Engine thực hiện biên dịch trang View này.

Khi sử dụng Razor View Engine, lập trình viên không cần phải học một ngôn ngữ lập trình mới, mà thay vào đó Razor cung cấp các mẫu nhằm kết hợp giữa cú pháp của ngôn ngữ đánh dấu siêu văn bản (HTML) và ngôn ngữ lập trình như C#, VB ở nội dung của các trang View. Với các lập trình viên đã quen với các ngôn ngữ C# hoặc VB thì học và làm quen với Razor View Engine là rất đơn giản.

#### 2.6.2.2. Cú pháp Razor

* Để thông dịch các đoạn mã được viết bởi các ngôn ngữ như C#, VB trong các trang View, Razor đưa ra những quy tắc nhằm để xác định các đoạn mã đó được biên dịch ở phía Server (server-side code) và các đoạn mã đánh dấu (markup code) được thực hiện biên dịch ở phía client (như javascript, css, html, …), Razor sử dụng ký tự @ nhằm phân biệt giữa server-side code và markup code.
* Khi tạo ra các Razor view chúng ta cần chú ý như sau :
* Sử dụng @{ code } để viết một khối mã lệnh
* Để viết một biểu thức inline (Viết biểu thức Razor trực tiếp trong đoạn mã của markup code, lập trình viên chỉ cần sử dụng @.
* Sử dụng từ khóa var để khai báo biến
* Sử dụng ; để kết thúc một câu lệnh
* Với các Razor view sử dụng ngôn ngữ C#, các file có đuôi mở rộng là .cshtml Với các Razor view sử dụng ngôn ngữ VB, các file có đuôi mở rộng là .vbhtml

#### 2.6.2.2. Khối mã lệnh

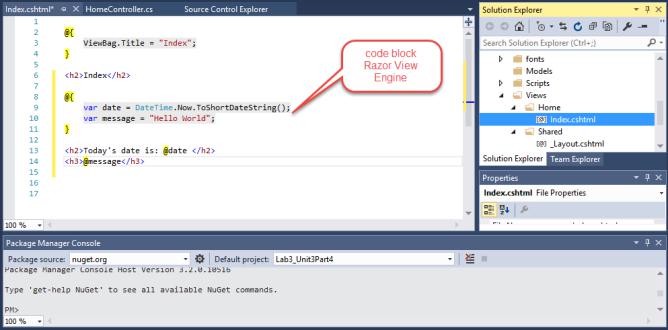
Razor hỗ trợ lập trình viên viết các khối mã lệnh tại các View. Một khối mã lệnh Raror là một phần của View, và nó được viết bằng các ngôn ngữ C# hoặc VB.

*Cú pháp @{*

*<code>*

*}*

*Mô tả*

* *code* : là các đoạn mã code được viết bởi ngôn ngữ C# hoặc VB và được biên dịch ở phía Server

*Hình 6. Minh họa khối mã lệnh Razor View*

### 2.6.3. CSS (Cascading Style Sheets)

CSS được viết tắc của cụm từ Cascading Style Sheets, hiểu đơn giản nó là tổng hợp các chuỗi dạng text, mỗi câu lệnh của CSS dùng để định dạng một phần tử của HTML về màu sắc, kích thước, font chữ...Nhờ có CSS mà giao diện của web, nội dung trên web trở nên rõ ràng hơn, hấp dẫn hơn với người đọc



*Hình 7. Vai trò CSS v ới Website*

### 2.6.4. Bootstrap

Bootstrap hiện đang là Framwork hỗ trợ tốt nhất và phổ biến nhất trong việc phát triễn các dự án về giao diện website cho cả thiết bị di động và máy tính bao gồm HTML, CSS và Java script. Bootstrap phát triễn qua nhiều phiên bản, phiên bản mới nhất lúc mình thực hiện tutorials là bản 3.3.2.

Bootstrap được sử dụng rộng rải bởi lợi ích to lớn mà nó mang lại như giao diện website tương thích với các thiết bị smartphone, tablet và máy tính thông thường vì có thể tự động co giản kích thước. Bootstrap được xây dựng với các module hỗ trợ sẵn nên rất dễ dàng tìm hiểu, xây dựng giao diện một cách nhanh chóng…

### 2.6.5. JavaScript, JQuery và Ajax

#### 2.6.5.1. JavaScript

JavaScript (có phần mở rộng .js) là một ngôn ngữ lập trình dạng kịch bản dựatrên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ được sử dụng rộng rãi trên các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng đối tượng có sẵn trong các ứng dụng. Giống Java, JavaScript có cú pháp tương tự C, nhưng gần giống với Self hơn Java. Hiện nay, có rất nhiều framework, libraries được viết bằng ngôn ngữ JavaScript:

* **jQuery**: Một thư viện mạnh mẽ và thông dụng.
* **AngularJS**: Một thư viện xây dựng ứng dụng Single Page.
* **NodeJS**: Một thư viện được phát triển phía Server dùng để xây dựng ứng dụng realtime.
* **Firebase**: Một nền tảng di động dùng lưu trữ dữ liệu theo thời gian thực

#### 2.6.5.2. Jquery

* jQuery là một thư viện kiểu mới của JavaScript, được tạo bởi John Resig vàonăm 2006 với một phương châm tuyệt vời: **Write less, do more**.
* jQuery làm đơn giản hóa việc truyền tải HTML, xử lý sự kiện, tạo hiệu ứng động và tương tác Ajax. Với jQuery, khái niệm Rapid Web Development đã không còn quá xa lạ.
* jQuery là một bộ công cụ tiện ích JavaScript làm đơn giản hóa các tác vụ đa dạng với việc viết ít code hơn. Dưới đây liệt kê một số tính năng tối quan trọng được hỗ trợ bởi jQuery:
* **Thao tác DOM** − jQuery giúp dễ dàng lựa chọn các phần tử DOM để traverse một cách dễ dàng như sử dụng CSS, và chỉnh sửa nội dung của chúng bởi sử dụng phương tiện Selector mã nguồn mở, mà được gọi là Sizzle. jQuery giúp tương tác với người dùng tốt hơn bằng việc xử lý các sự kiện đa dạng mà không làm cho HTML code rối tung lên với các Event Handler
* **Hỗ trợ AJAX** − jQuery giúp bạn rất nhiều để phát triển một site giàu tính năng và phản hồi tốt bởi sử dụng công nghệ AJAX.
* **Hiệu ứng** − jQuery đi kèm với rất nhiều các hiệu ứng đa dạng và đẹp mắt mà bạn có thể sử dụng trong các Website của mình.
* **Gọn nhẹ** − jQuery là thư viện gọn nhẹ - nó chỉ có kích cỡ khoảng 19KB (gzipped).
* **Được hỗ trợ hầu hết bởi các trình duyệt hiện đại** − jQuery được hỗ trợ
* hầu hết bởi các trình duyệt hiện đại, và làm việc tốt trên IE 6.0+, FF 2.0+, Safari 3.0+, Chrome và Opera 9.0+
* **Cập nhật và hỗ trợ các công nghệ mới nhất** − jQuery hỗ trợ CSS3 Selector và cú pháp XPath cơ bản.

#### 2.6.5.3. Ajax

AJAX, là viết tắt của Asynchronous JavaScript and XML, (tạm dịch là JavaScript và XML không đồng bộ), là một kỹ thuật mới để tạo các ứng dụng web giàu tính tương tác, nhanh hơn và mượt mà hơn với sự giúp đỡ của XML, HTML, CSS và JavaScript.

AJAX không phải là ngôn ngữ lập trình mới. Tên AJAX ở đây không có nghĩa là bạn phải học XML trước khi cần học AJAX. Điều này là không cần thiết.

AJAX cho phép các trang web được cập nhật một cách không đồng bộ bằng cách trao đổi các lượng dữ liệu nhỏ với Server. Tức là AJAX giúp cập nhật các phần nhỏ trong trang mà không cần tải lại toàn bộ trang. Nếu đến đây bạn chưa hiểu thì các dòng tiếp theo đây sẽ giúp bạn hiểu cách AJAX làm việc hơn.

### 2.6.6. Phương thức mã hóa dữ liệu MD5

Viết tắt của Message-Digest algorithm 5 (Giải thuật Tiêu hóa tin 5) là một hàm băm mật mã học được diễn tả bằng một số hệ thập lục phân 32 ký tự, theo chuẩn RFC 1321, các chương trình MD5 thường được gọi là MD5Sum. Nó được dùng để tạo ra một chuỗi 128 bit duy nhất từ một chuỗi dữ liệu nhập bất kỳ, và vì thế thường được dùng để kiểm tra tính toàn vẹn dữ liệu của một tập tin

# CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1. XÁC ĐỊNH ACTOR

### 3.1.1. Quản trị viên hệ thống

* Là thành viên có cấp độ cao nhất khi giao tiếp với hệ thống với toàn quyền thao tác với hệ thống.

### 3.1.2. Người dùng hệ thống (khách hàng)

* Thao tác trên giao diện bán hàng và có quyền truy cập đầy đủ các năng trên trang bán.

## 3.2. XÁC ĐỊNH USECASE

### 3.2.1. Quản trị viên hệ thống

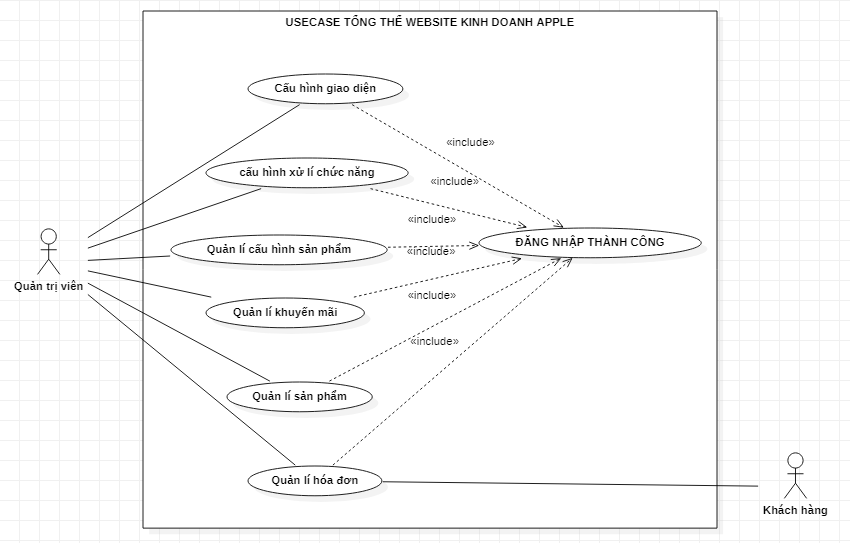
* Thay đổi cấu trúc giao diện hệ thống
* Quản trị tài khoản khách hàng
* Quản trị csdl sản phẩm
* Quản trị đơn hàng

### 3.2.2. Người dùng hệ thống

* Quản trị đơn đặt đơn hàng

## 3.3. MÔ HÌNH TƯƠNG TÁC NGƯỜI DÙNG VÀ HỆ THỐNG

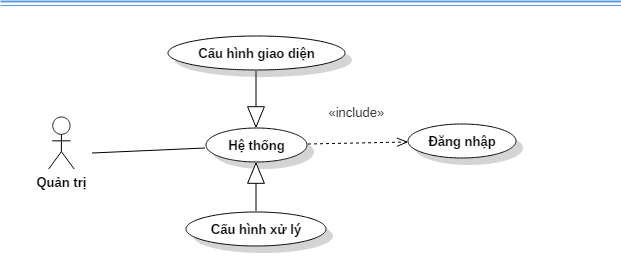
### 3.3.1. Sơ đồ tổng thể



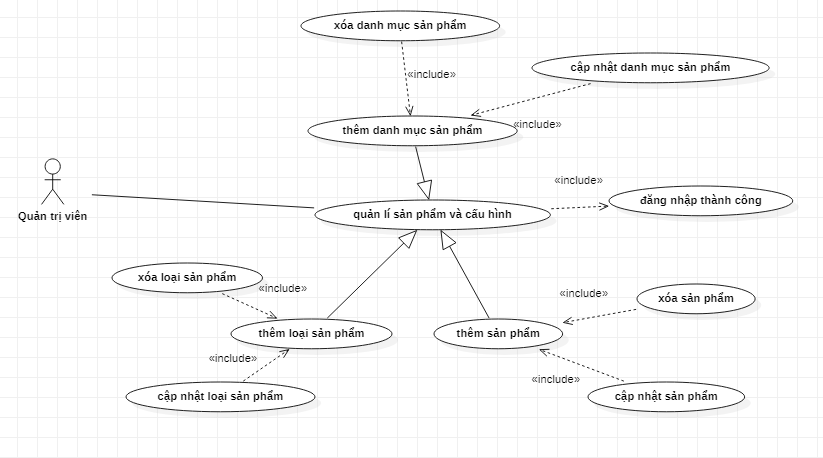
### 3.3.2. Sơ đồ phân rã Actor

#### 3.3.2.1 Actor quản trị viên

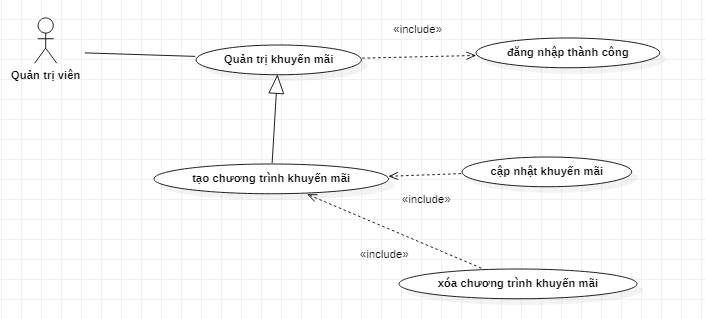
##### 3.3.2.1.1 Cấu hình giao diện



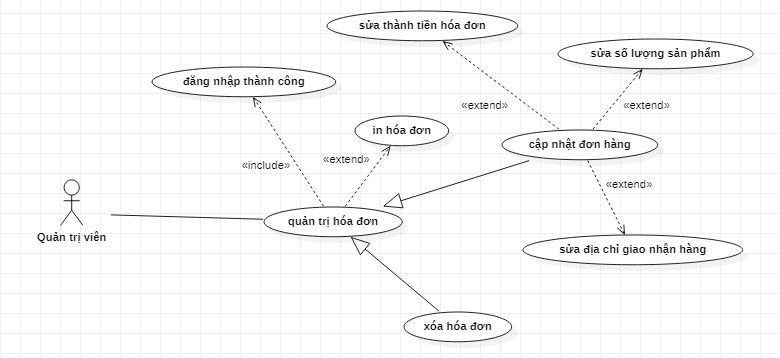
##### 3.3.2.1.2 Quản lí hóa đơn và Cấu hình sản phẩm



##### 3.3.2.1.3 Quản lí khuyến mãi

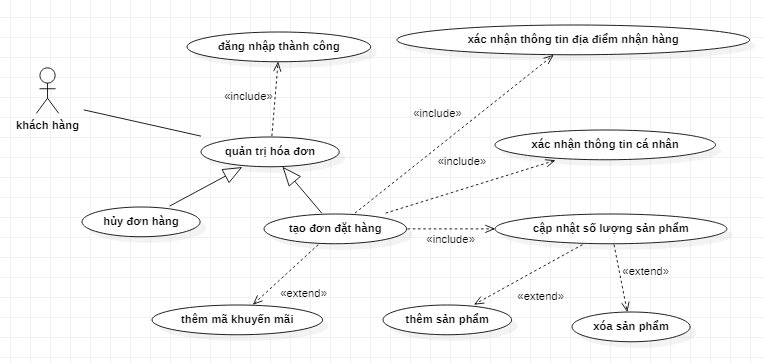


##### 3.3.2.1.4 Quản lí hóa đơn



#### 3.3.2.2 Actor khách hàng

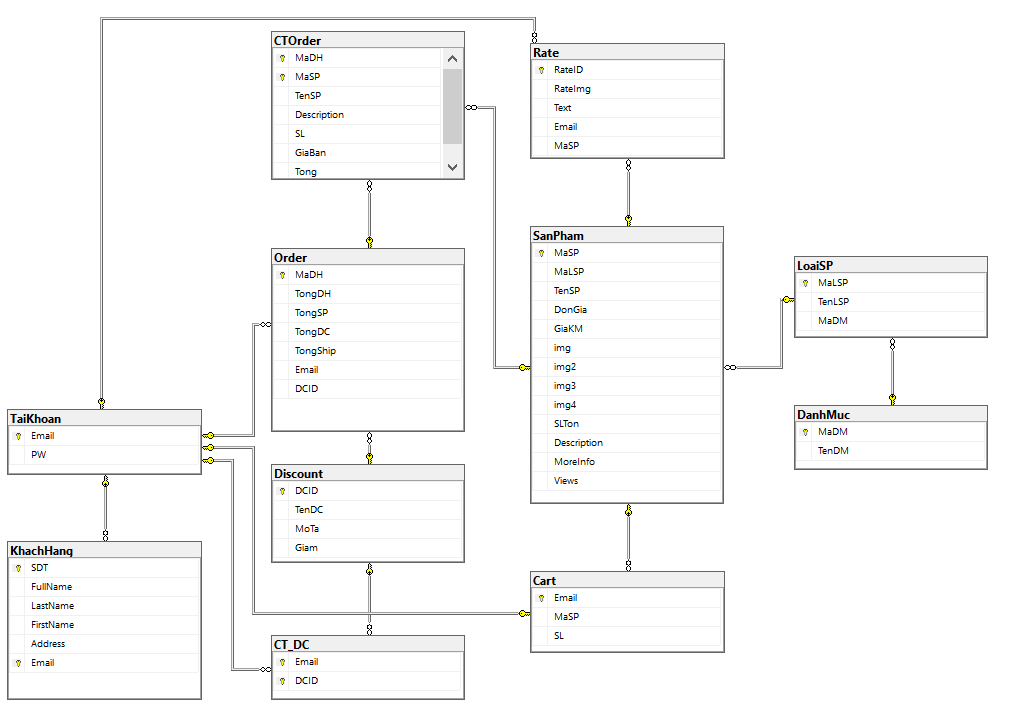
##### 3.3.2.2.1 Quản lí hóa đơn



#### 3.3.2.1.Actor Quản trị viên

### 3.3.3. Đặc tả các UseCase tổng quát

## 3.4. MÔ HÌNH CƠ SỞ DỮ LIỆU



*Hình 3.4 Mô Hình Cơ Sở Dữ Liệu*

## 3.5. Tổ chức bảng dữ liệu

### 3.5.1 Tài Khoản (Taikhoan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | **Email** | **varchar(50)** | **Not Null** | **Email** |
| 2 | PW | varchar(20) |  | Password |

### 3.5.2 Khách Hàng (Khachhang)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | **SDT** | **varchar(10)** | **Not Null** | **Số điện thoại** |
| 2 | FullName | nvarchar(50) |  | Họ và tên |
| 3 | LastName | nvarchar(20) |  | Họ |
| 4 | FirstName | nvarchar(30) |  | Tên |
| 5 | Address | nvarchar(MAX) |  | Địa chỉ |
| 6 | Email | varchar(50) | Not Null | Email |

### 3.5.3 Danh Mục (Danhmuc)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | **MaDM** | **varchar(10)** | **Not Null** | **Mã danh mục** |
| 2 | TenDM | nvarchar(50) |  | Tên danh mục |

### 3.5.4 Loại Sản Phẩm (LoaiSP)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | **MaLSP** | **varchar(15)** | **Not Null** | **Mã loại sản phẩm** |
| 2 | TenLSP | nvarchar(50) |  | Tên loại sản phẩm |
| 3 | MaDM | varchar(10) |  | Mã danh mục |

### 3.5.5 Sản Phẩm (SanPham)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | **MaSP** | **varchar(10)** | **Not Null** | **Mã sản phẩm** |
| 2 | MaLSP | varchar(15) |  | Mã loại sản phẩm |
| 3 | TenSP | nvarchar(50) |  | Tên sản phẩm |
| 4 | DonGia | float |  | Đơn giá |
| 5 | GiaKM | float |  | Giá khuyến mãi |
| 6 | img | varchar(MAX) |  | ảnh |
| 7 | img2 | varchar(MAX) |  | ảnh 2 |
| 8 | img3 | varchar(MAX) |  | ảnh 3 |
| 9 | img4 | varchar(MAX) |  | ảnh 4 |
| 10 | SLTon | smallint |  | Số lượng tồn |
| 11 | Description | nvarchar(MAX) |  | Mô tả |
| 12 | MoreInfo | nvarchar(MAX) |  | Thêm thông tin |
| 13 | Views | int |  | Lượt xem |
| 14 | LoaiHang | smallint |  | Loại hàng |

### 3.5.6 Đơn Hàng (Order)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | **MaDH** | **varchar(10)** | **Not Null** | **Mã đơn hàng** |
| 2 | TongDH | float |  | Tổng đơn hàng |
| 3 | TongSP | float |  | Tổng sản phẩm |
| 4 | TongDC | float |  | Tổng khuyến mãi |
| 5 | TongShip | float |  |  |
| 6 | Email | varchar(50) |  | Email |
| 7 | DCID | varchar(10) |  | Mã khuyến mãi |

### 3.5.7 Chi Tiết Đơn Hàng (CT\_Order)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | MaDH | varchar(10) | Not Null | Mã đơn hàng |
| 2 | MaSP | varchar(10) | Not Null | Mã sản phẩm |
| 3 | TenSP | nvarchar(50) |  | Tên sản phẩm |
| 4 | Description | varchar(50) |  | Mô tả |
| 5 | SL | smallint |  | Số lượng |
| 6 | GiaBan | float |  | Giá bán |
| 7 | Tong | float |  | Tổng |

### 3.5.8 Đánh Giá Sản Phẩm (Rate)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | **RateID** | **varchar(10)** | **Not Null** | **Mã đánh giá** |
| 2 | RateImg | varchar(50) |  | ảnh đánh giá |
| 3 | Text | nvarchar(MAX) |  | Đánh giá |
| 4 | Email | varchar(50) |  | Email |
| 5 | MaSP | varchar(10) |  | Mã sản phẩm |

### 3.5.9 Giỏ Hàng (Cart)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | Email | varchar(50) | Not Null | Email |
| 2 | MaSP | varchar(10) |  | Mã sản phẩm |
| 3 | SL | smallint |  | Số lượng |

### 3.5.10 Khuyến Mãi (Discount)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | **DCID** | **varchar(10)** | **Not Null** | **Mã khuyến mãi** |
| 2 | TenDC | nvarchar(50) |  | Tên khuyến mãi |
| 3 | MoTa | nvarchar(MAX) |  | Mô tả |
| 4 | Giam | float |  | Giảm |

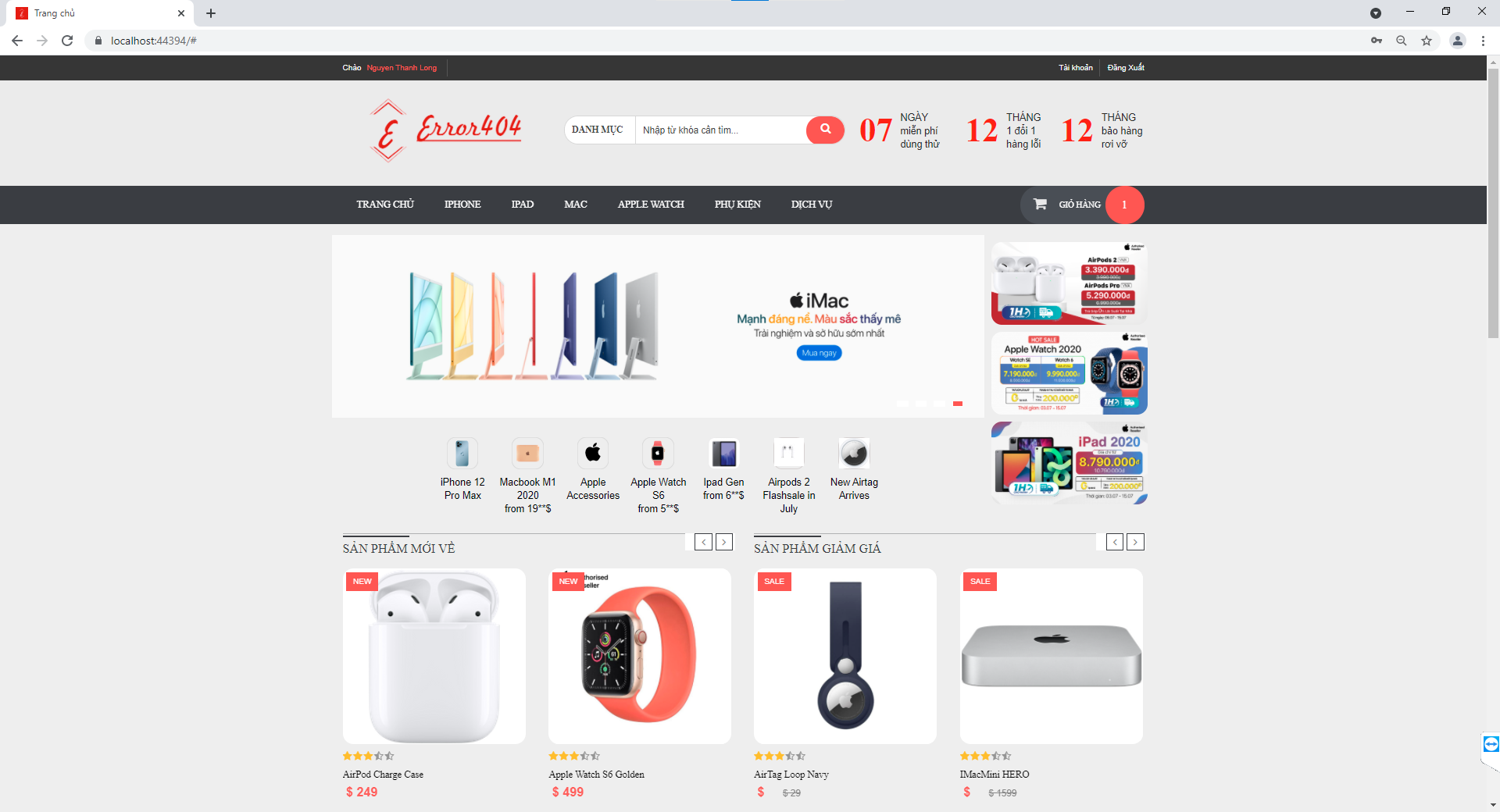
### 3.5.11 Chi Tiết Khuyến Mãi(CT\_DC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | TÊN CỘT | KIỂU DỮ LIỆU | NOT NULL | DIỄN GIẢI |
| 1 | Email | varchar(50) | Not Null | email |
| 2 | DCID | varchar(10) |  | Mã khuyến mãi |

# CHƯƠNG IV: DEMO VÀ TRIỂN KHAI HỆ THỐNG

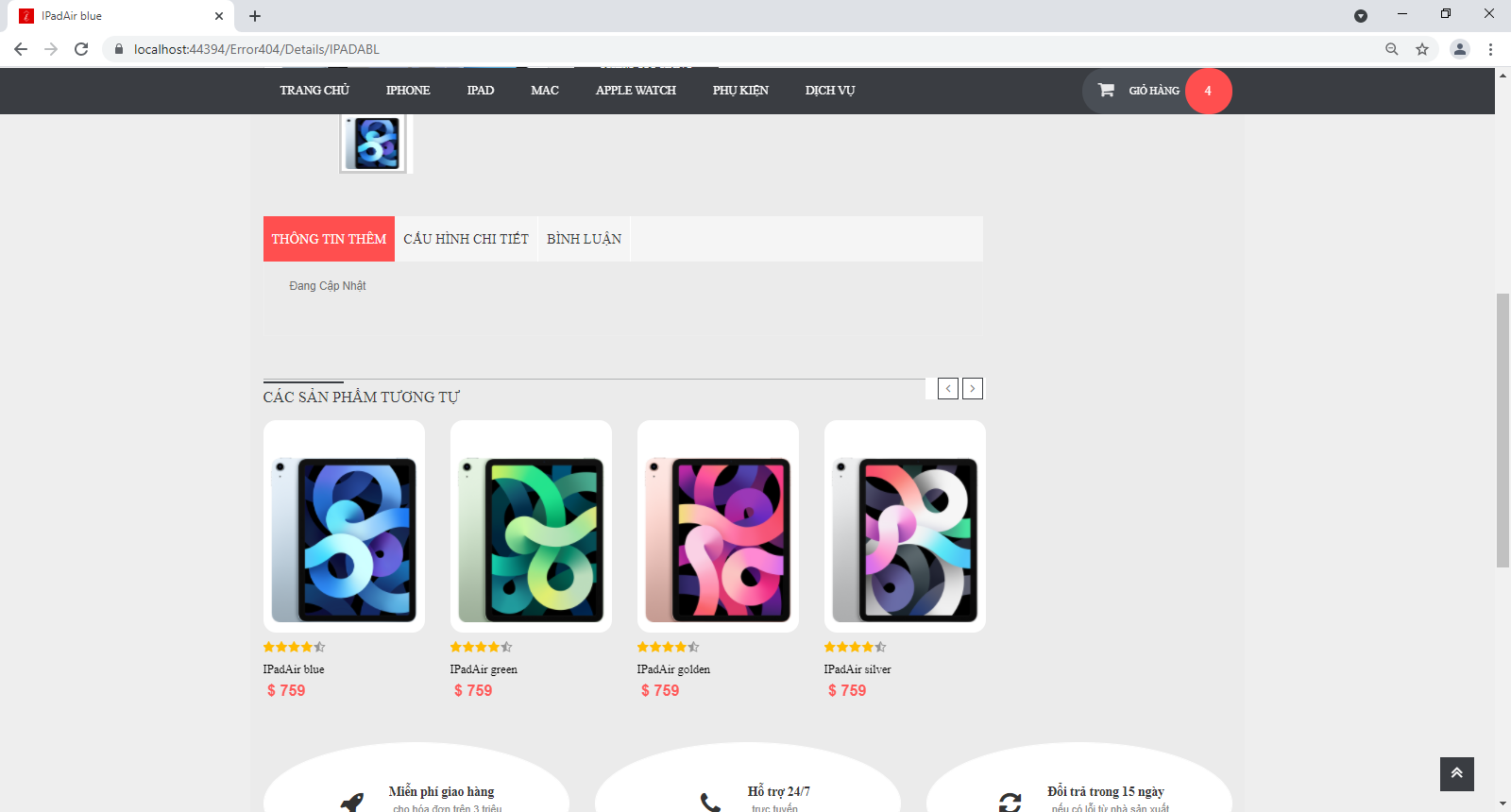
## 4.1. Giao diện chức năng:

### 4.1.1 Giao diện chính:



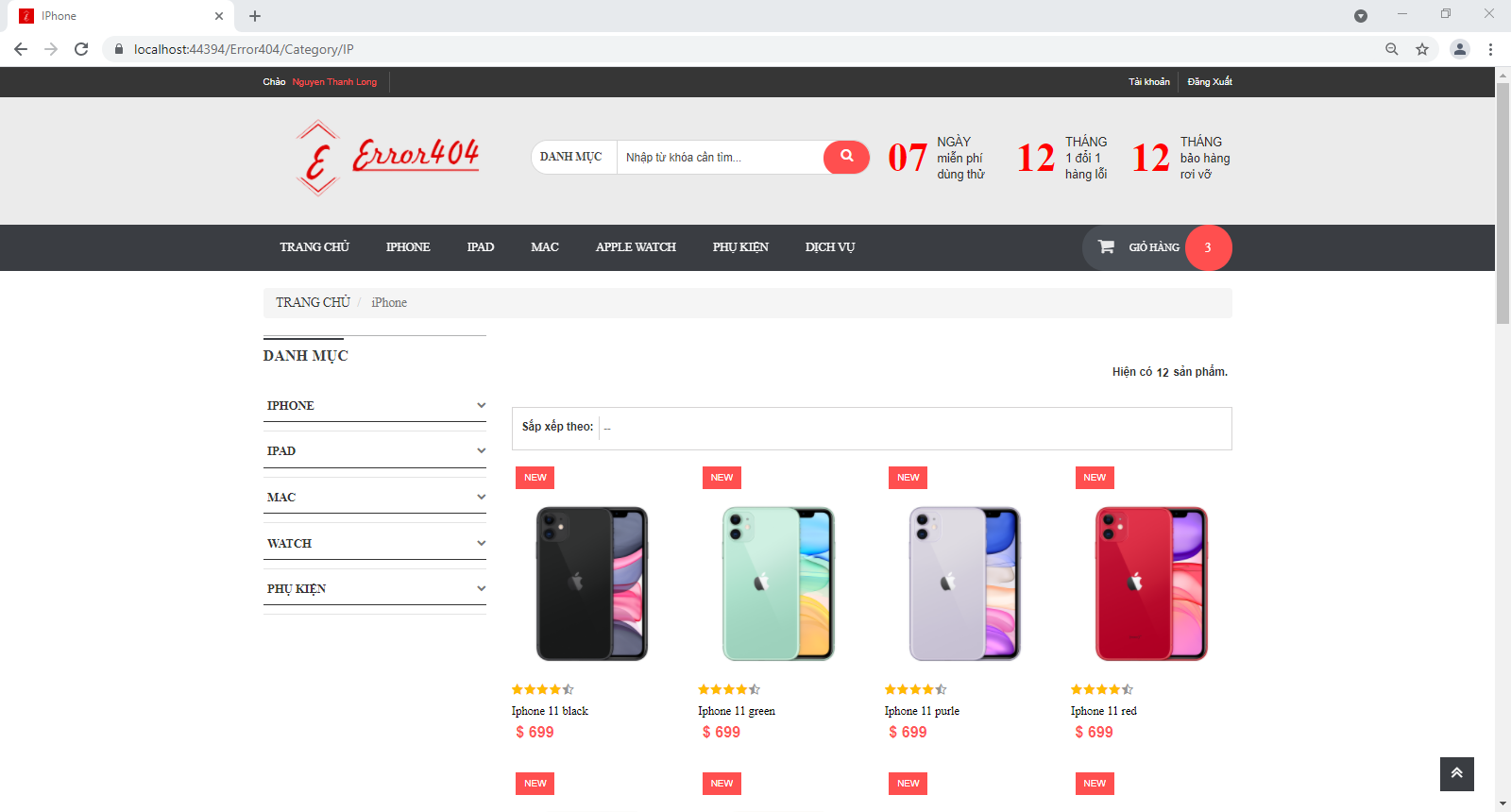
Giao diện chính của website được hiển thị với đầy đủ các trang liên kết sau khi quá trình đăng nhập vào hệ thống được diễn ra thành công.

### 4.1.2 giao diện đơn sản phẩm:



Giao diện đơn sản phẩm của website được hiển thị với đầy đủ các thông tin chi tiết của một sản phẩm khi click vào một sản phẩm nhất định.

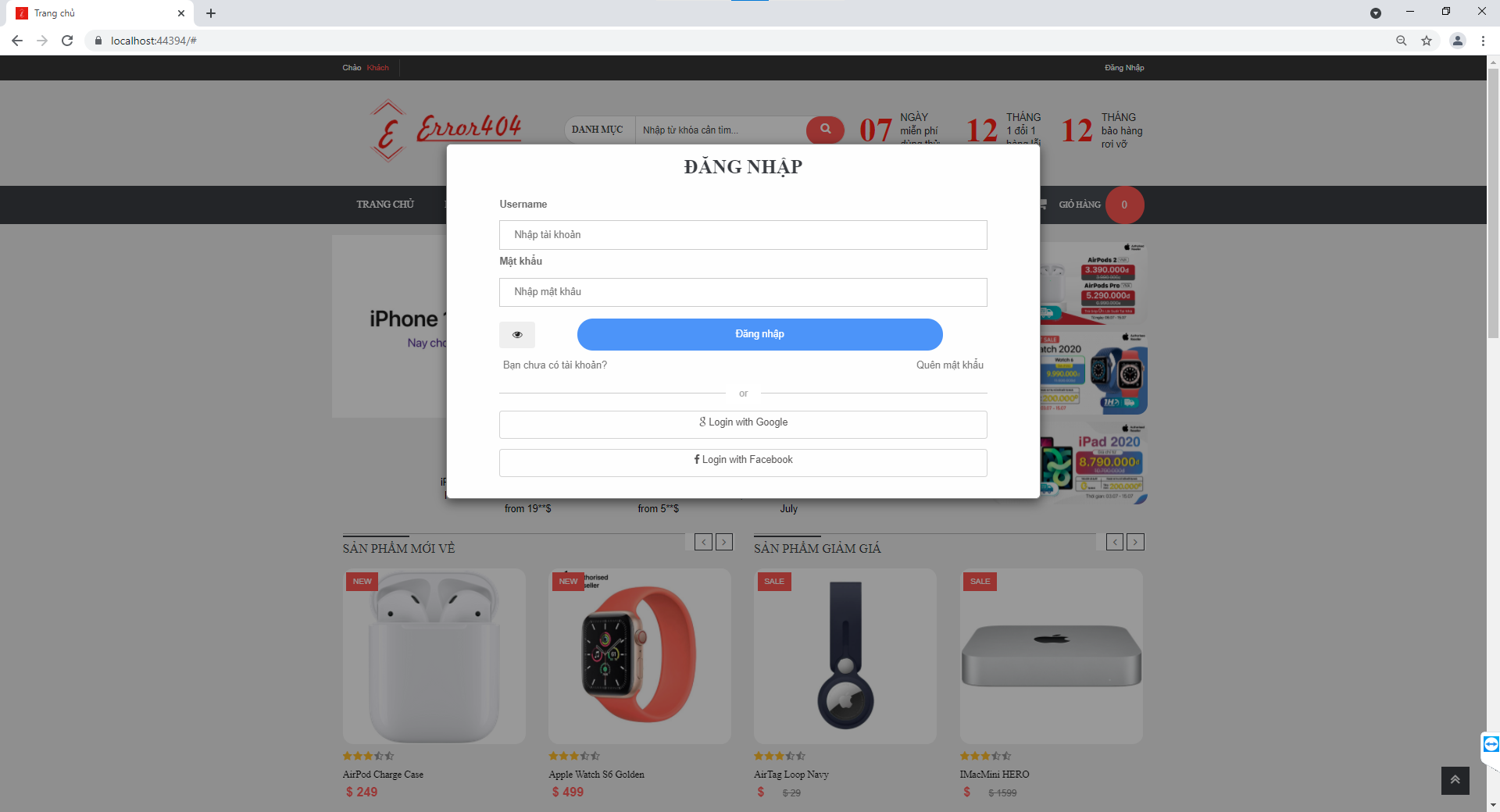
### 4.1.3 giao diện danh mục sản phẩm:



Giao diện danh mục của website được hiển thị với đầy đủ các thông tin chi tiết của một danh mục sản phẩm khi click vào một danh mục sản phẩm nhất định

### 4.1.4 giao diện chức năng:

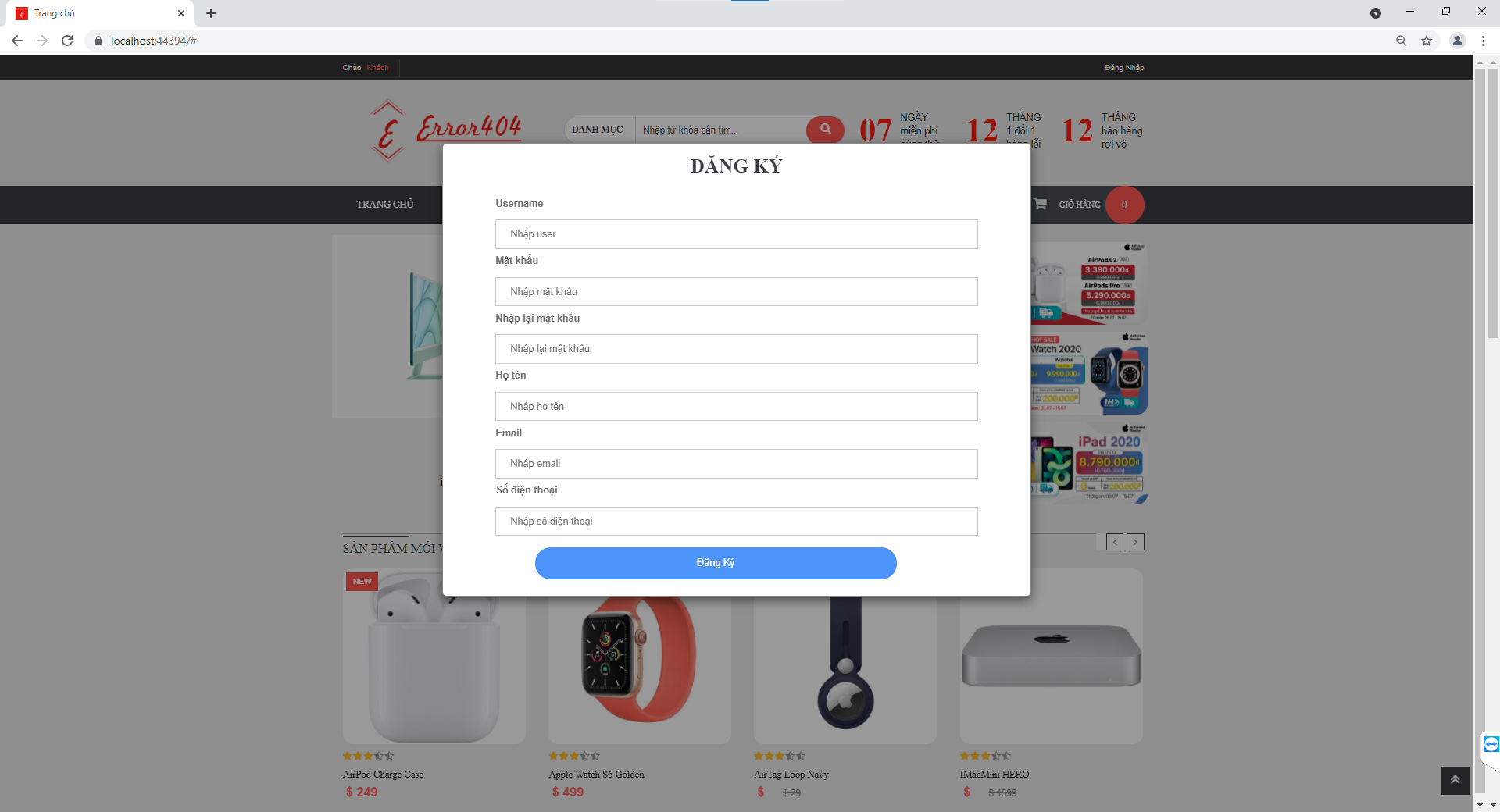
#### 4.1.4.1 giao diện chức năng đăng nhập:



Khi click vào button đăng nhập, View đăng nhập được hiển thị với yêu cầu nhập [UserName] và [Mật khẩu] để được truy cập vào hệ thống.

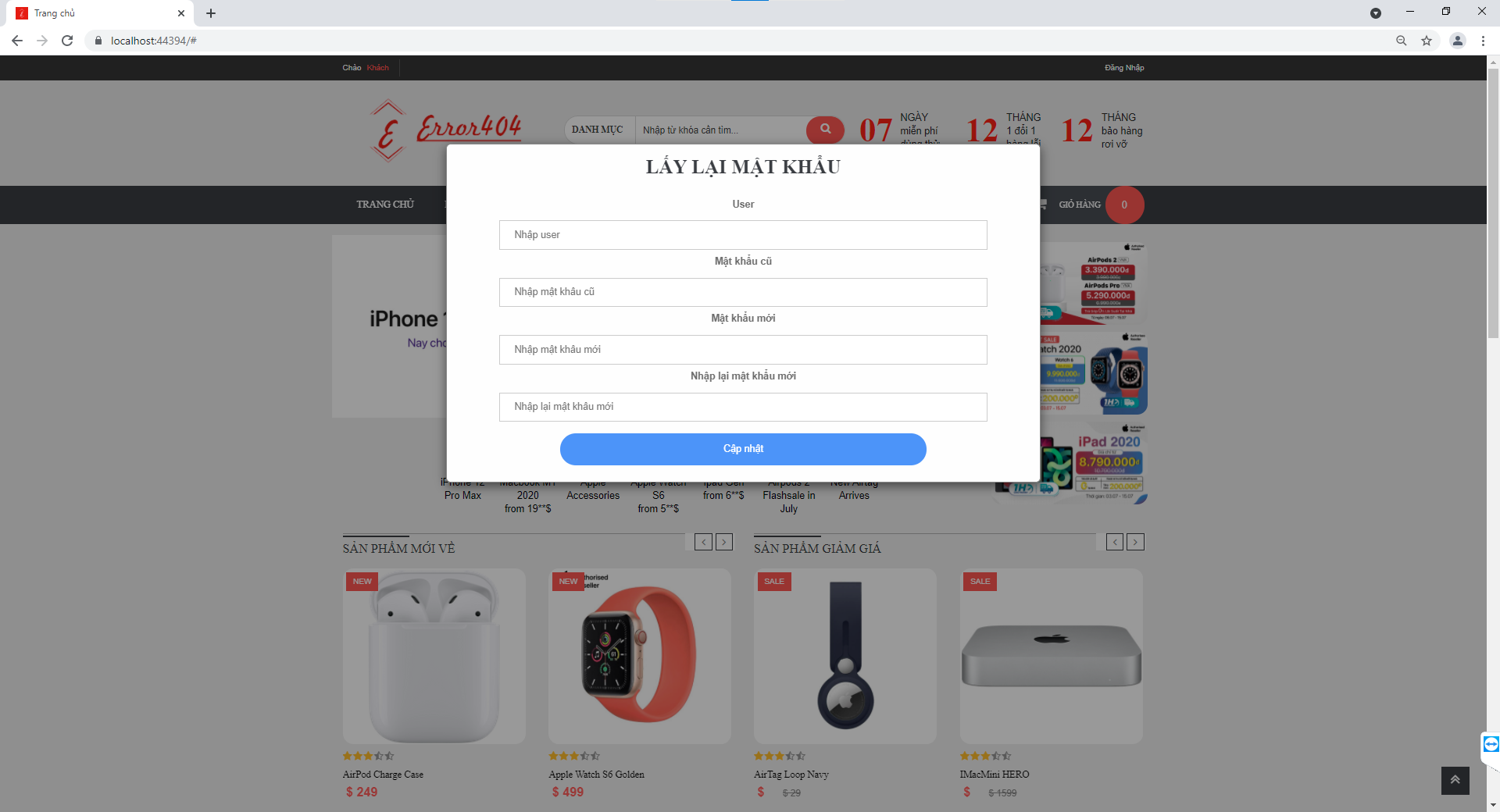
Nếu nhập sai thông tin, hệ thống sẽ phát thông báo “Đăng nhập thất bại, vui lòng thử đăng nhập lại”. Và quá trình cứ được lặp lại cho đến khi được nhập đúng để truy cập vào hệ thống hoặc người dùng chủ động thoát khỏi hệ thống.

#### 4.1.4.2 giao diện chức năng tạo tài khoản:



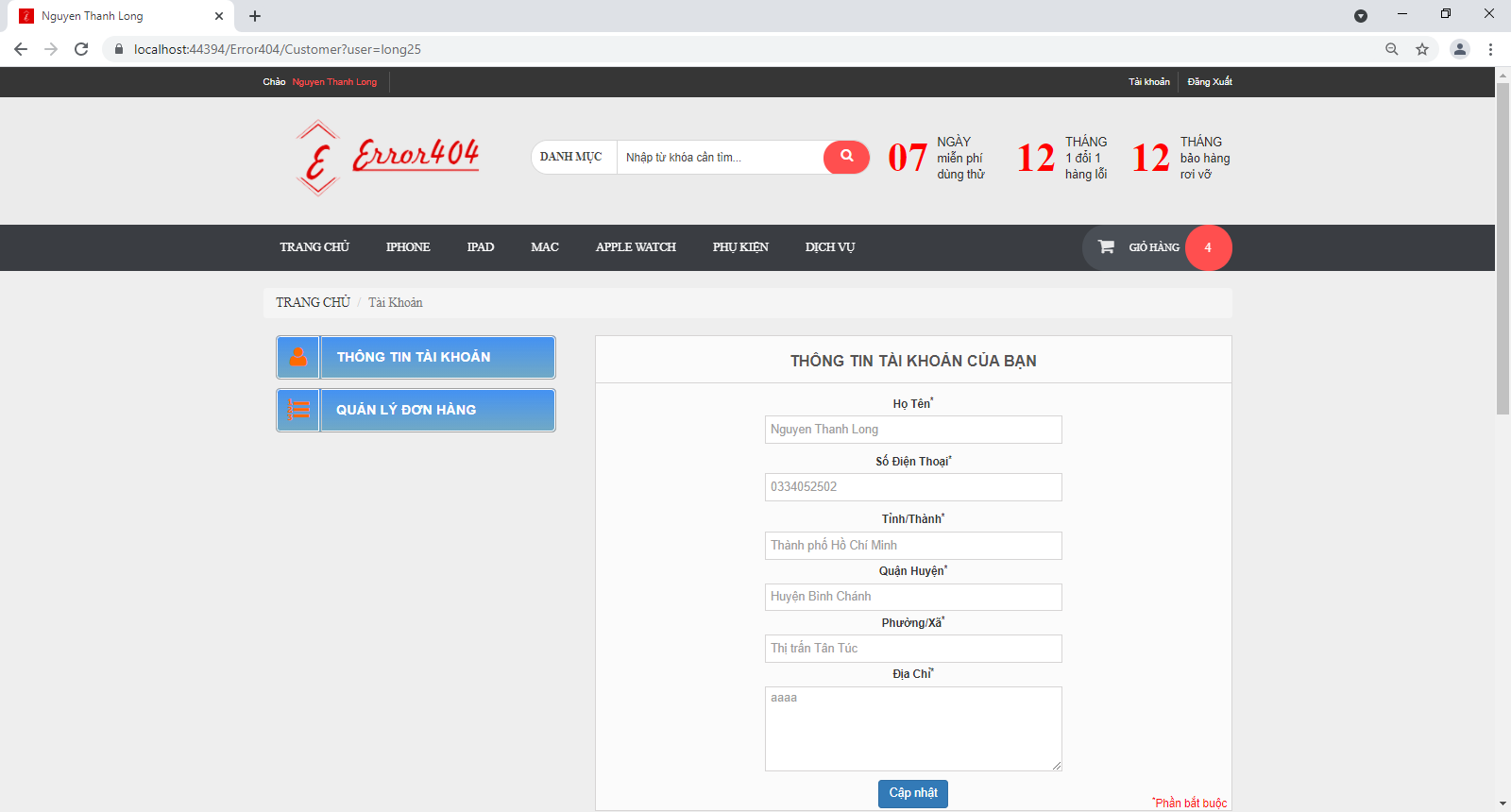
Khi click vào button đăng kí hoặc button chưa có tài khoản, view đăng kí hiển thị yêu cầu nhập username, mật khẩu, nhập lại mật khẩu, họ tên, email, số điện thoại để đăng kí tài khoản.

#### 4.1.4.3 giao diện chức năng quên mật khẩu:

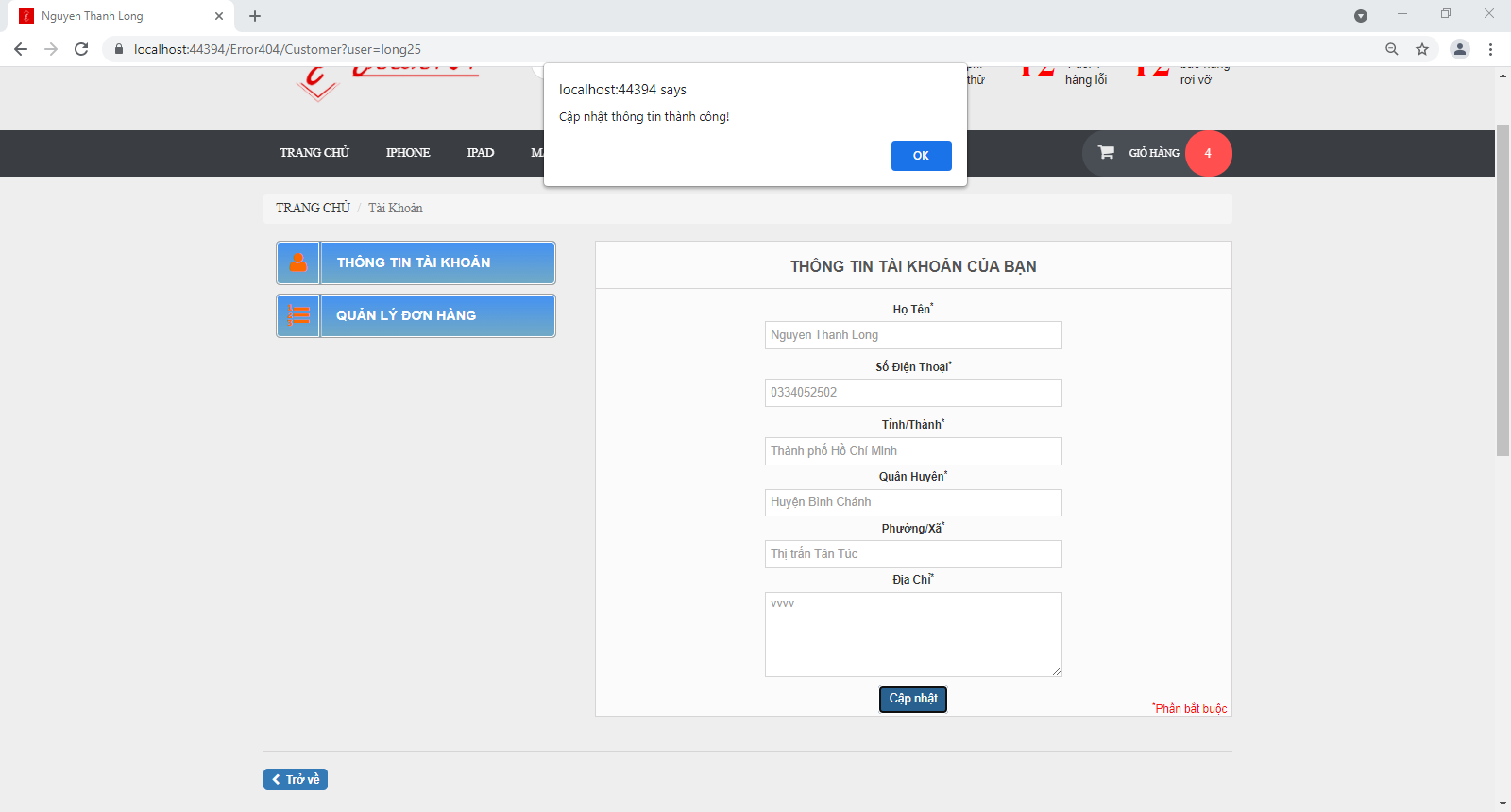


Khi click vào link quên mật khẩu, view quên mật khẩu hiện thị yêu cầu nhập username hoặc sdt để nhận mã xác thực lấy lại mật khẩu.

#### 4.1.4.4 giao diện chức năng cập nhật thông tin tài khoản:

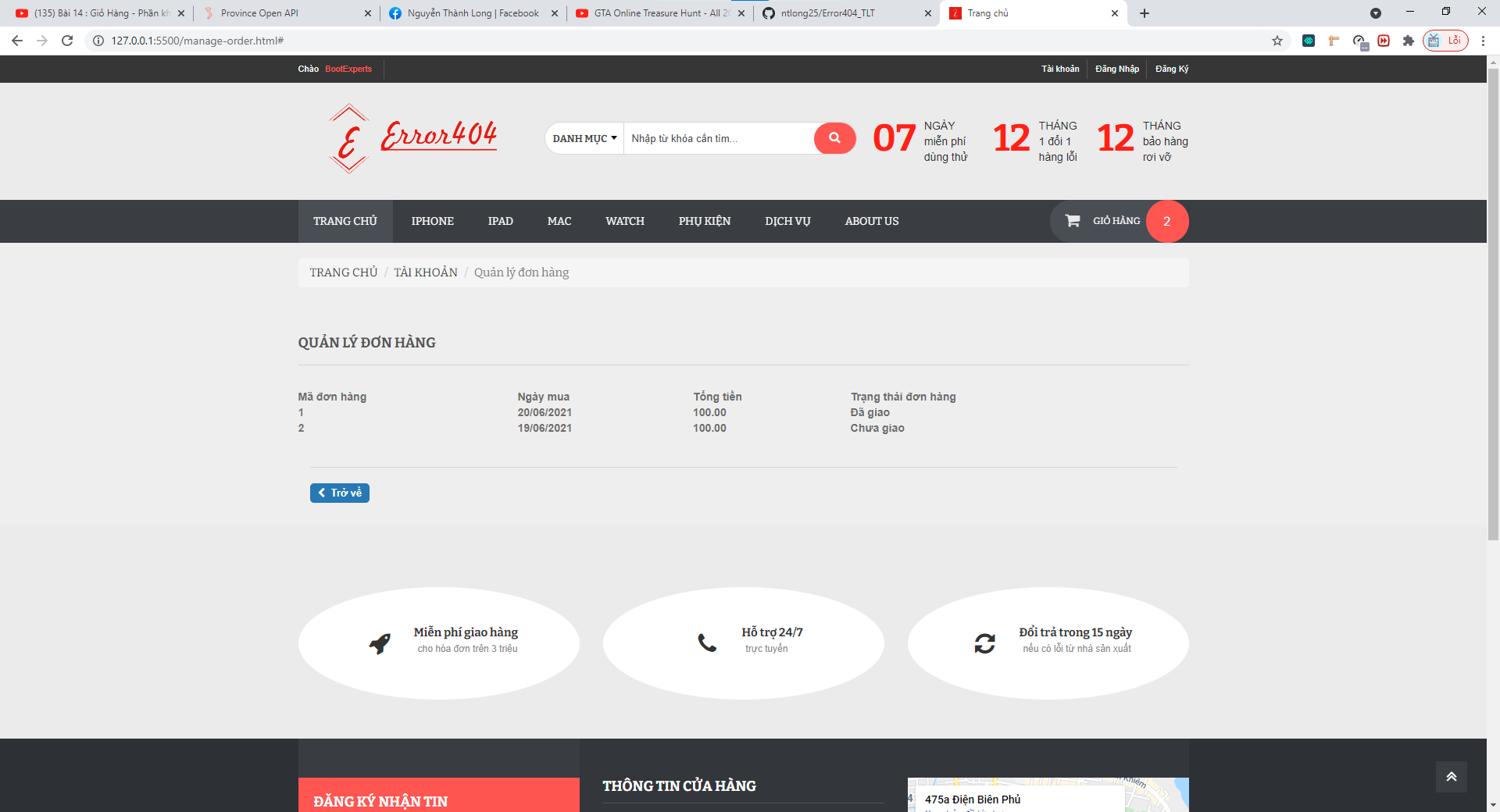


Trong link tài khoản, vào thông tin tài khoản của bạn để cập thông tin.

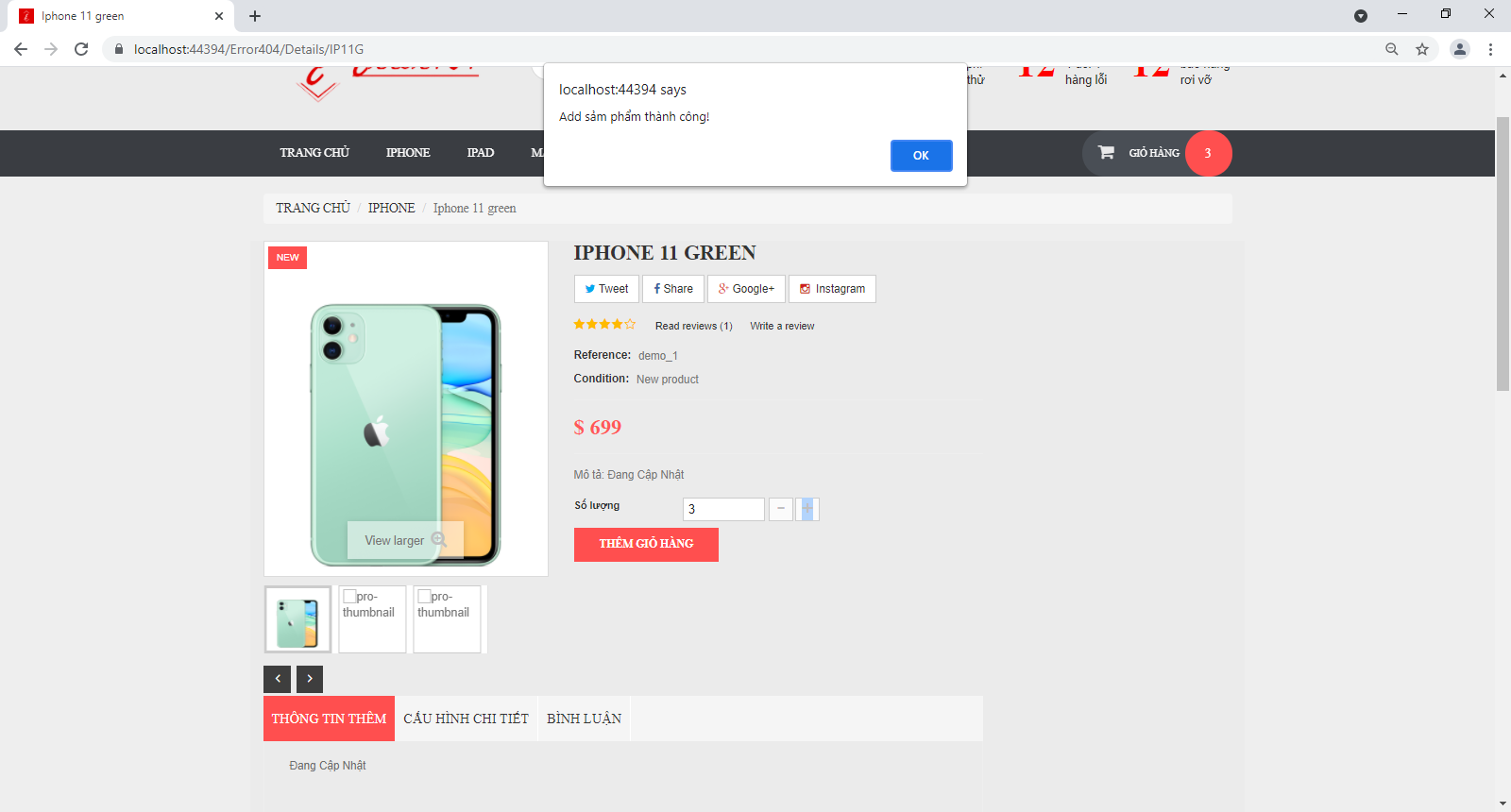


Thông báo hiển thị đã cập nhật thông tin thành công khi nhấn button cập nhật.

#### 4.1.4.5 GIAO DIỆN CHỨC NĂNG QUẢN LÝ ĐƠN HÀNG

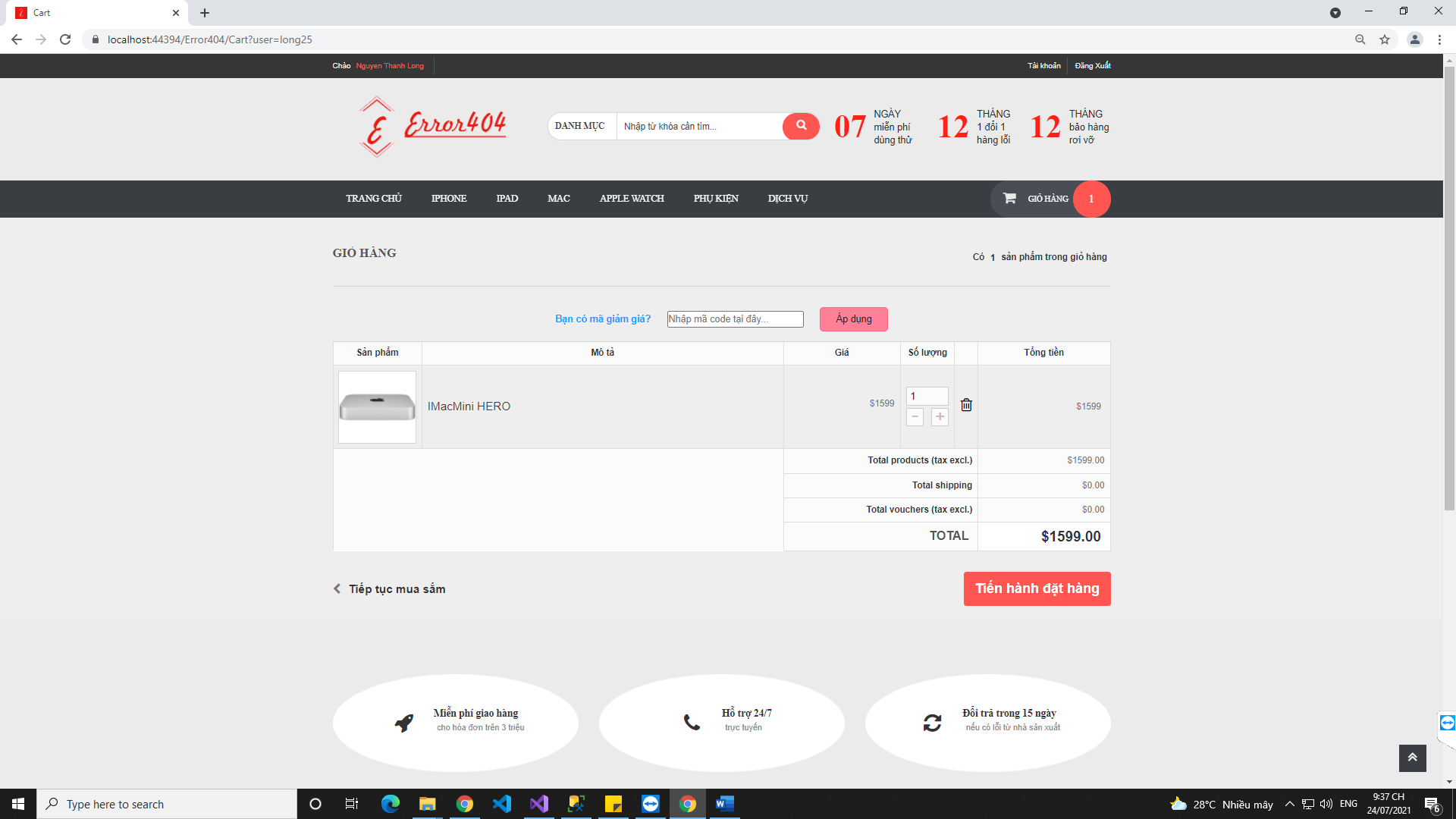


#### 4.1.4.6 giao diện chức năng đặt hàng:

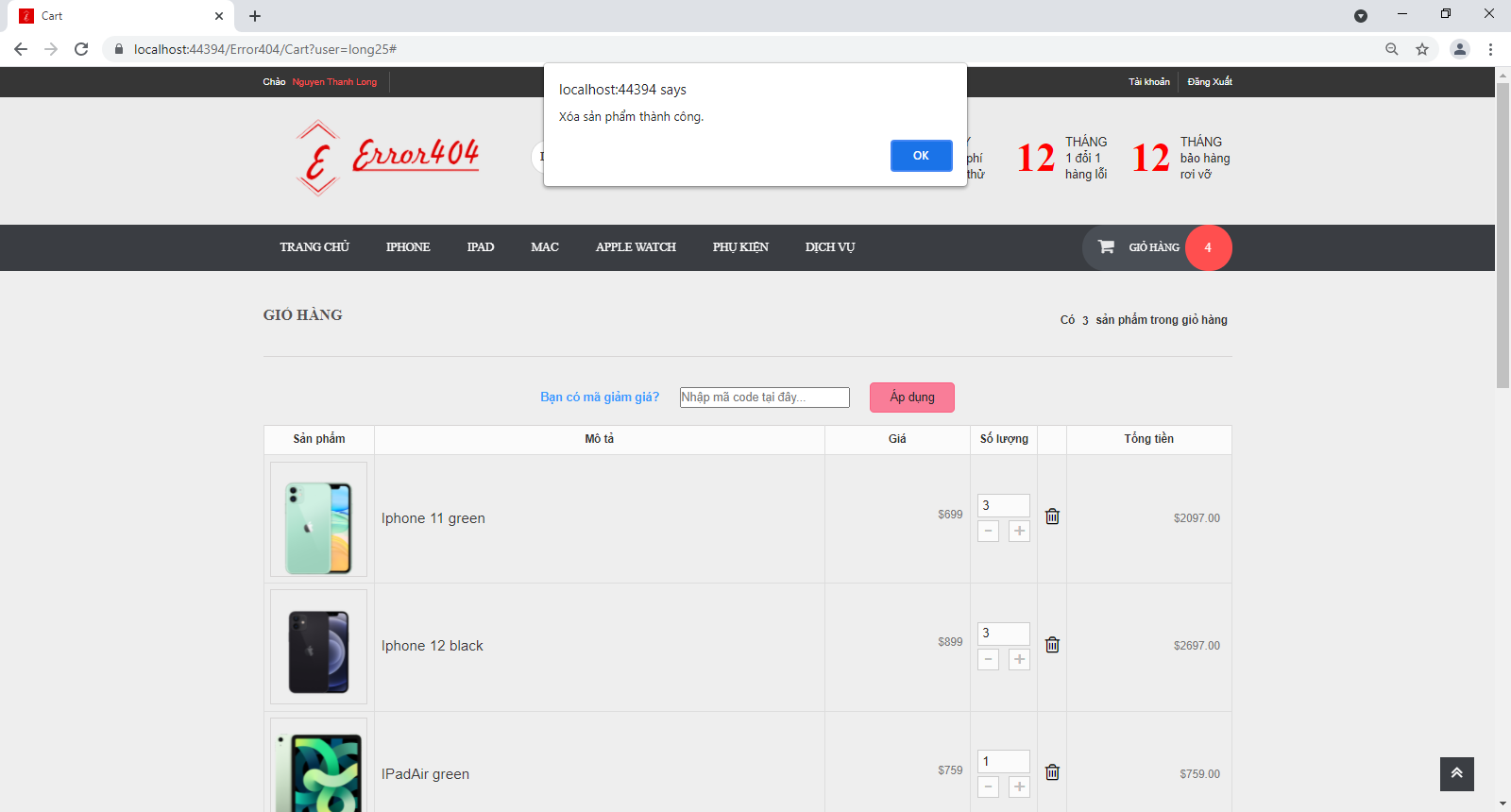


Thêm sản phẩm vào giỏ hàng bằng ấn vào button giỏ hàng hoặc icon giỏ hàng khác trên website.

Trang web sẽ hiện thị thông báo đã thêm sản phẩm vào giỏ hàng thành công.

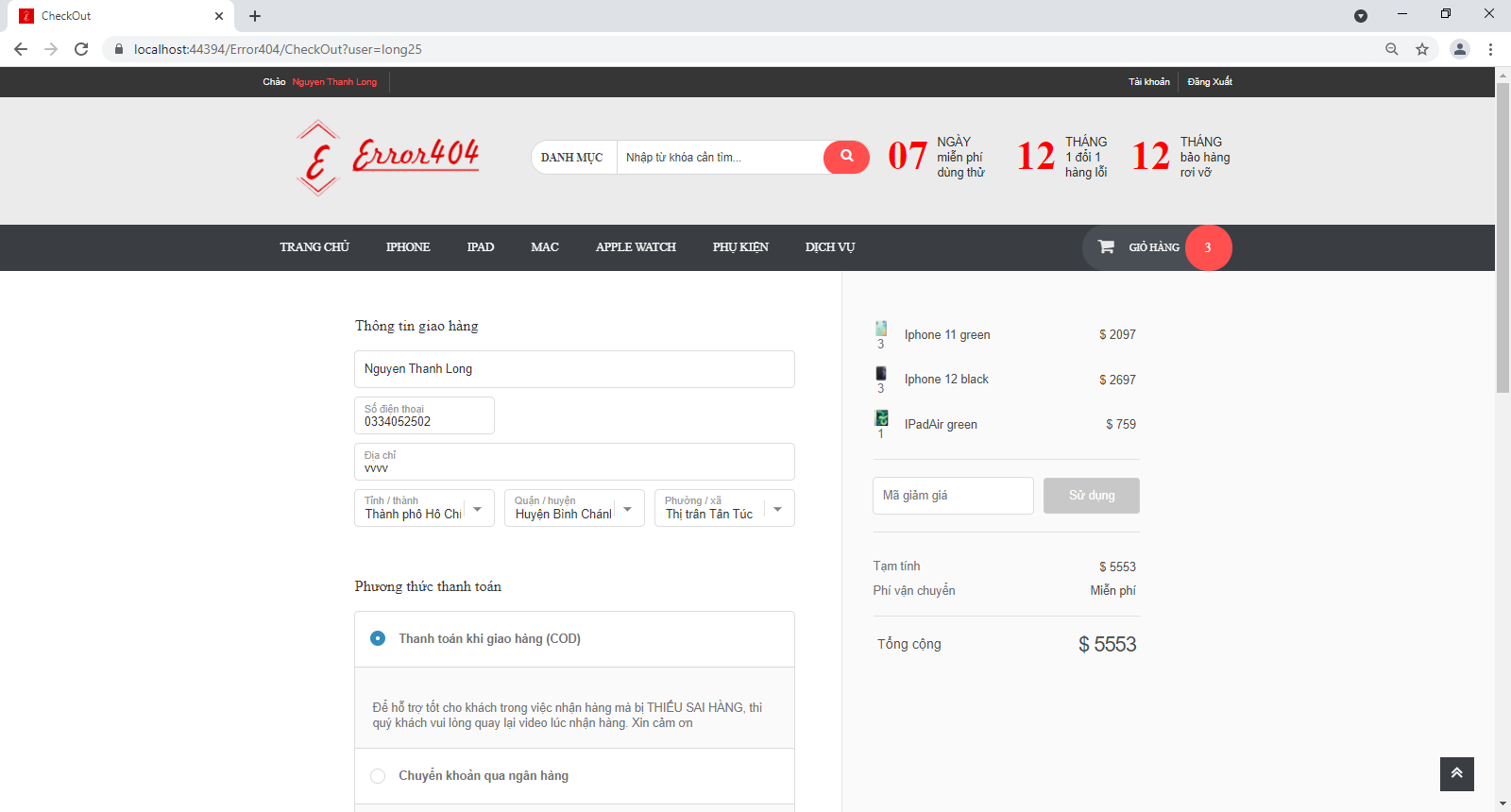


Giao diện trang web sẽ chuyển qua trang thông tin sản phẩm đã thêm vào giỏ hàng khi nhấn nút thanh toán.



Có thể thêm nhiều sản phẩm vào giỏ hàng.

Giao diện trang web hiển thị thông báo đã xóa sản phẩm thành công khi nhấn vào nút xóa sản phẩm



Giao diện trang web chuyển sang trang xác nhận lại các thông tin để nhận hàng chọn các đơn vị vận chuyển, phương thức thanh toán cũng như thêm mã giảm giá vào đơn hàng trước khi đặt hàng thành công.

# CHƯƠNG V: KẾT LUẬN

## 5.1. KẾT QUẢ VÀ KINH NGHIỆM

### 5.1.1. Kết quả

* Đã xây dựng được website bán hàng với nền tảng ASP.NET MVC5 với đầy đủ các tính năng cần thiết.
* Ứng dụng công nghệ Bootstrap vào hệ thống, góp phần tăng tính tương thích về cấu trúc giao diện cho nhiều kích thước màng hình khác nhau, từ điện thoại đến máy tỉnh bảng và Laptop/PC.
* Ứng dụng được các công nghệ JavaScript, Json, Ajax,…

### 5.1.2. Kinh nghiệm

* Rèn luyện kỹ năng tự học và giải quyết vấn đề.
* Củng cố kiến thức về: lập trình web (HTML, CSS, JavaScript), ASP.NET C#, LINQ, ADO.NET, Sql Server.

## 5.2. HẠN CHẾ

* Kiến thức về MVC còn hạn chế
* Phương thức code và tổ chức code chưa được tối ưu
* Thời gian thực hiện còn hạn chế
* Chưa nhận được sự giúp đỡ hay hướng dẫn từ giảng viên

## 5.3. HƯỚNG PHÁT TRIỂN

* Tích hợp vào hệ thống các API của các sàn thương mại điện tử như Lazada,Shopee…
* Nâng cao tốc độ thực thi và hiệu suất của hệ thống.
* Nâng cao tính năng bảo mật của hệ thống
* Nâng cấp hệ thống tự tính phí vận chuyển trên toàn nước
* Nâng cấp và phát triển một trang tích hợp quản lí bán hàng trực tuyến đồng bộ với website bán hàng.

# Tài liệu tham khảo

* <https://www.tutorialsteacher.com/mvc/asp.net-mvc-tutorials>
* <https://www.w3schools.com/>
* Giáo trình môn Lập Trình Web đại học HUTECH