|  |  |
| --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**  **KHOA CÔNG NGHỆ THÔNG TIN**  Description: C:\Documents and Settings\Administrator\Desktop\logo dai hoc_khong nen.png  **ĐỒ ÁN MÔN HỌC**  LẬP TRÌNH WEB  **Giảng viên giảng dạy : Nguyễn Mai Huy**  **Bùi Duy Tân**  **Sinh viên thực hiện : Huỳnh Thị Hồng Ngân**  **MSSV : 1911546868**  **Chuyên ngành : Kỹ thuật phần mềm**  **Môn học : Lập trình web**  **Khóa : 2019**  **Tp.HCM, tháng 05 Năm 2021** | |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **KHOA CÔNG NGHỆ THÔNG TIN**  🙜 🙜 🙝 | | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập – Tự do – Hạnh phúc**  🙜 🙜 🙝 | |

**LỜI NÓI ĐẦU**

Ứng dụng công nghệ thông tin ở thời nay ngày càng được áp dụng nhiều vào đời sống hằng ngày và là một trong những yếu tố quan trọng không thể thiếu trong cuộc sống hằng ngày. Và nó có tầm quan trọng không nhỏ trong mọi lĩnh xung quanh ta như kinh tế, quân sự, quốc phòng,…

Ứng dụng công nghệ thông tin giúp chúng ta giải quyết nhanh và tiện lợi các vấn đề trong cuộc sống hơn. Và Internet ngày càng phát triển mạnh mẽ và đã trở thành nhu cầu không thể thiếu trong cuộc sống của chúng ta. Nó là công cụ truyền tải tạo ra những giá trị lớn lao và giúp nâng cao chất lượng cuộc sống của chúng ta hơn. Lấy ví dụ như ngày nay các đồ thiết bị gia dụng trong gia đình của chúng ta hầu hết đều được hỗ trợ bởi các công nghệ tiên tiến. Như máy giặt ngày càng hiện đại có thể điều khiển từ xa được việc giạt và hẹn giờ tắt. Tivi, tủ lạnh có thể cảm ứng vân tay mang lại nhiều trải nghiệm hấp dẫn,… Công nghệ được tích hợp trong việc giảng dạy và giáo dục giúp cho giáo viên có thể truyền tải thông tin một cách cặn kẽ và dễ dàng cho học sinh dễ nắm bắt kiến thức nhanh hơn. Với tình hình dịch bệnh thì việc học online cũng được các trường áp dụng vào việc giảng dạy làm cho học sinh không bị trì hoãn kiến thức,.. Ứng dụng công nghệ trong việc kinh doanh mang lại việc các đơn hàng được lưu thông nhanh chóng giúp cho người bán và người mua dễ dàng chọn và trao đổi trực tiếp trên các trang web.,.. Ngoài ra còn nhiều mặt khác trong cuộc sống mà chúng ta đều nhờ công nghệ.

Hoạt động thương mại điện tử ngày nay ngày càng được người tiêu dùng chọn vì nó đơn giản và tiện lợi rất nhiều giúp chúng ta được kiệm được khá nhiều thời gian. Đối với các nhà kinh doanh thì việc giới thiệu các sản phẩm tới khách hàng một cách nhanh chóng. Vì vậy các Website bán hàng ra đời nhằm mục đích đưa sản phẩm của mình tới người tiêu dùng để dễ dàng chọn lựa mang lại nhiều hiệu quả cho việc tang doanh thu. Các nhà cung ứng, nhà kinh doanh sẽ nhận được số tiền mà mình đã bán hàng qua các website đó.

Trong thời gian thực hiện đồ của môn lập trình web em đã được sự giúp đỡ nhiệt tình, những lời khuyên bổ ích, và sự đóng góp nhiệt tình của thầy cô và bạn bè. Em xin chân thành cảm ơn sự hướng dẫn trực tiếp của giảng viên thầy Nguyễn Mai Huy và thầy Bùi Duy Tân đã giảng dạy và chỉ bảo em tận tâm trong quá trình học tập. Các thầy luôn giải đáp thắc mắc và đưa ra lời khuyên giúp em hiểu được vấn đề của môn học.

Sự giảng dạy nhiệt tình và luôn giải đáp tạo mọi tình huống cho sinh viên chúng em có thể hiểu được bài một cách nhanh chóng. Ở các thầy em học được tính tự giác để giải quyết các vấn đề liên quan đến bài học. Vấn đề nào chư ahieuer có thể tựu học trên mạng hoặc có thể tham khảo qua một vài tài liệu do các thầy cung cấp. Em xin chân thành cảm ơn quý thầy cô trong khoa Công nghệ Thông tin đã tận tình giảng dạy, trang bị cho em những kiến thức quý báu trong năm vừa qua.

Con xin chân thành cảm ơn ông bà, cha mẹ đã luôn động viên ủng hộ vật chất lẫn tinh thần trong suốt thời gian qua. Có được chỗ dựa vững chắc từ phía gia đình luôn ủng hộ sự đam mê của con. Qua việc làm đồ án lập trình web cũng giúp em hiểu được sự thực tế hơn để sau này có thể áp dụng vào việc làm.

Em Xin chân thành cảm ơn!

Sinh viên thực hiện:Huỳnh Thị Hồng Ngân

Lớp:19DTH2B

Tp Hồ Chí Minh, ngày tháng 4, năm 2021

# 

MỤC LỤC

**[PHẦN I: LÝ THUYẾT](#_Toc72791478)** [5](#_Toc72791478)

**[Chương I: Các kĩ thuật- công nghệ liên quan được sử dụng trong sản phẩm](#_Toc72791479)** [5](#_Toc72791479)

**[Chương II)ASP.NET VÀ MVC DESIGN PATTERN](#_Toc72791480)** [5](#_Toc72791480)

**[1)Mô hình MVC là gì?](#_Toc72791481)** [6](#_Toc72791481)

**[4)ASP.NET MVC là gì ?](#_Toc72791482)** [8](#_Toc72791482)

**[7)Tạo mới project ASP.NET MVC](#_Toc72791483)** [9](#_Toc72791483)

**[II) ASP.NET VÀ MVC DESIGN PATTERN](#_Toc72791484)** [10](#_Toc72791484)

**[III)Layout for View in ASP.Net](#_Toc72791485)** [11](#_Toc72791485)

**[IV)ENTITY fRAMWORK](#_Toc72791486)** [14](#_Toc72791486)

**[(1)ORM?](#_Toc72791487)** [14](#_Toc72791487)

**[V)LINQ](#_Toc72791488)** [17](#_Toc72791488)

**[VI)STORING AND TRACKING](#_Toc72791489)** [20](#_Toc72791489)

**[(1)IMPLICIT OBJECT](#_Toc72791490)** [20](#_Toc72791490)

**[(2)REQUEST OBJECT – HTTPREQUESTBASE CLASS](#_Toc72791491)** [20](#_Toc72791491)

**[(3) RESPONSE OBJECT](#_Toc72791492)** [21](#_Toc72791492)

**[PHẦN II- KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG](#_Toc72791493)** [25](#_Toc72791493)

**[I-PHÂN TÍCH YÊU CẦU ĐỀ](#_Toc72791494)** [25](#_Toc72791494)

**[2.YÊU CẦU ĐẶT RA](#_Toc72791495)** [25](#_Toc72791495)

**[PHẦN III- PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG](#_Toc72791496)** [26](#_Toc72791496)

**[1)KHẢO SÁT HIỆN TRẠNG VÀ XÂY DỰNG HỆ THỐNG](#_Toc72791497)** [26](#_Toc72791497)

**[2)Công cụ thực hiện](#_Toc72791498)** [35](#_Toc72791498)

**[PHẦN IV:TÀI LIỆU THAM KHẢO](#_Toc72791499)** [35](#_Toc72791499)

**DANH MỤC CÁC HÌNH**

Hinh1: sơ đồ của MVC

Hinh2: Cú pháp của phần Layout về RenderBodey() cần có

Hinh3: LinQ gồm có một số công dụng

Hinh4:Danh scahs các bảng trong database cảu MySQL

Hinh5: trang chủ Home/Index của User

Hinh6: trang chủ admin

Hinh7:trang login của admin

Hinh8: tạo snar phẩm mới của admin

Hinh9: danh sách chi tiết của các sản phẩm

Hinh10:giao diện đăng kí của user

Hinh11:Giao diện đăng nhập của user

Hinh12:Trang hiển thị dnah mục của các sản phẩm

Hinh13: trang chi tiết về 1 snar phẩm

Hinh14: sơ đồ phân tích về chọn và đặt hàng

Hinh15: sơ đồ thanh toán

Hinh16:sơ đò đăng kí thành viên(User**)**

**Danh mục các từ ngữ viết tắt**

MVC(Model-Control-View)

HTML(Hypertext Markup Language)

ORM(Object Relational Mapping)

DB(database)

URL(Uniform Resource Locator)

# **PHẦN I: LÝ THUYẾT**

**Chương I: Các kĩ thuật- công nghệ liên quan được sử dụng trong sản phẩm**

**1)Mô hình Client- Server:**

+Là mô hình gồm 2 thành phần là máy khách (client) và máy chủ (server). Server là nơi lưu trữ tài nguyên để thực hiện yêu cầu của máy khách (client). Client thì gồm máy tính và thiết bị điện tử nói chung.

+Client-Sever cho phép người dùng chung mạng tập trung và chức năng một số dịch vụ file chuyên dụng

+Cho phép người dùng chia sẻ đồng thời cùng một tài nguyên

+Mô hình Client-Server là mô hình nổi tiếng được dùng rộng rãi và được dùng phổ biến trong các trang web hiện nay. Ngoài ra còn có mô hình master-slaver thì máy chủ sẽ gửi dữ liệu đến máy con dù cho máy con có cần hay không

**2)Ưu điểm :**

+Có khả năng chống quá tải mạng

+Đảm bảo toàn vẹn dữ liệu khi có sự cố xảy ra

+Dễ dàng mở rộng hệ thống mạng

+Tổng hợp các kĩ thuật hiện đại: GIS, mô hình thiết kế hướng đối tượng,…

**3)Nhược điểm :**

+Càn bảo dưỡng, bảo trì Server thường xuyên

+Khả năng bảo mật thông tin còn hạn chế

**4)Về nguyên lý hoạt động**

Cả Server và Client đều dựa trên các giao thức chuẩn

+Về Client : là máy khách- máy trạm là nơi gửi yêu cầu đến Server. Khi Client tiếp nhận yêu cầu của người dùng sau đó sẽ gửi đến server. Khi server nhận được kết quả, client sẽ tổ chức và trình diễn những kết quả đó

+Về Server : sẽ xử lý yêu cầu của client. Sau khi xử lí xong thì server sẽ gửi trả kết quả

…..

**Chương II)ASP.NET VÀ MVC DESIGN PATTERN**

### **1)Mô hình MVC là gì?**

-MVC Design Pattern là một mẫu thiết kế nỏi tiếng dung để phát triển ứng dụng

-Các ngôn ngữ đều cho phép người dung MVC (năm 2010 cho phép như một framword)

-MVC là viết tắt của Model-View-Controller

-MVC là tên một mô hình về thiết kế cấu trúc thành phần Source Code của một dự án lập trình. Hay nói ngắn gọn là quy tắc tổ chức Source Code

-Model để chứa dữ liệu, View để trình bày dữ liệu và hiển thị giao diện nhận tương tác với người dùng, Controller để đọc và điều khiển dữ liệu ra View, đồng thời xử lí tương tác của người dùng

**1.1)Model :**

+Được xem là một mô hình dữ liệu và business logic. Có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng.

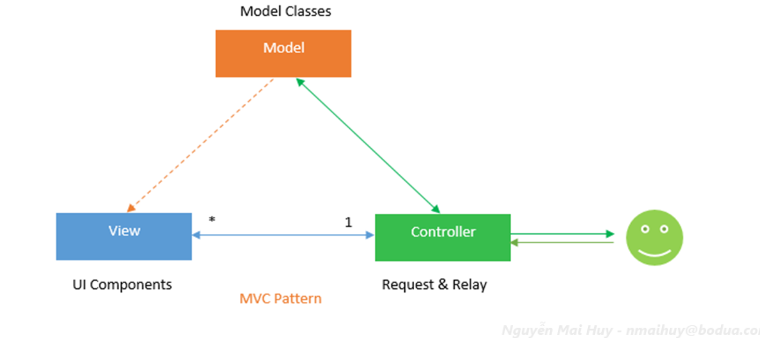
+Là cầu nối giữa View và Controller.

+Model thể hiện hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML. Model cho phép thể hiện các thao tác với cơ sở dữ liệu (xem, truy xuất, xử lý dữ liệu,…)

**1.2)View :**

+Là phần giao diện dành cho người dùng xem, client sẽ xem được thông tin dữ liệu MVC thông qua tìm kiếm hoặc sử dụng website

+View còn có chức năng ghi nhận thông tin người dùng để tương tác với Controller



*Hinh1*

**1.3)Controller :**

+Có nhiệm vụ xử lí các yêu cầu người dùng đưa đến thong qua view.

+Controller còn có chức năng kết nối với model

=>Mỗi bộ phận có từng chức năng riêng nhưng chúng lại có sự thống nhất. Và nó được tích hợp nhiều chức năng khác nhau

-Mẫu thiết kế mang tính đối tượng dữ liệu, presention information (trình bày thông tin) xem để hiển thị thông tin dẫn dắt người dùng và control information(điều khiển các thông tin đáp ứng những nhu cầu đểg phù hợp người dùng

\*Ưu và nhược điểm của mô hình MVC

-Ưu điểm:

+Giúp cho các nhà lập trình có thể làm việc đồng thời trên Model- View- Controller

+Mô hình này có nhiều chế độ xem

+Khung nhìn cho một mô hình cụ thể được nhóm lại với nhau

+MVC cho phép nhóm các hành động liên quan trên một bộ điều khiển với nhau

-Nhược điểm:

+Nó điều hướng vì những lớp này mới và trừu trượng, yê cầu người dùng phải thích ứng với các tiêu chí đặt ra

+Kiến thức về công nghệ được sử dụng nhiều

Mô hình cấu trúc lập trình Web:

**2)Trong lập trình web chúng ta chia 3 tầng:**

+Presentation Layer(tầng 3): tầng này giúp hiển thị giao diện

+Business Logic Layer(tầng 2): tầng này chứa các hàm thực thi, giúp xử lí sự kiện xảy ra trên tầng Presentation Layer

+Data Access Layer(tầng 1):tầng này chứa hàm kết nối với các hệ quản trị cơ sở dữ liệu như SQL Server, Mysql, Oracle,…

Theo như mô tả, cấu trúc cổ điển này trộn vào nhau. Khi làm việc với tầng này nó khó xử lý giao diện

**3)Nhược điểm khi sử dung Web Forms:**

+Webs Forms không có sự tách biệt rõ ràng 3 tầng trong mô hình lập trình web

+Trong Web Forms có sử dụng 1 đối tượng View State mà đối tượng này làm cho trang web bị tang kích thước vì vậy làm giảm hiệu năng ứng dụng

+Giao diện của web forms được thiết kế dựa trên những toolbox có sẵn mà web forms cung cấp mà hiện nay đa phần designer sử dụng html, css để thiết kế website vì nó gây khó khắn trong việc thiết kế webb hiện nay, việc thiết kế trở nên mất linh hoạt

=>Với những khó khăn thì ASP.net MVC ra đời

Tuy ra đời 2009 thì 2013 nó đã là MVC 5. Ngày càng phát triển mạnh mẽ và được cập nhật những phiên bản mới và những công nghệ mới

\*Ưu điểm và nhược điểm của mô hình MVC

### **4)ASP.NET MVC là gì ?**

+ASP.NET MVC là một framework web được phát triển bởi Microsoft, thực thi mô hình MVC(Model-View-Controller). Nền tảng này tuân thủ quy tắc của mô hình MVC. Sử dụng công nghệ .NET Framweok được viết bởi ngôn ngữ C#

+ASP.NET MVC phiên bản đầu tiên ra đời từ năm 2017. Đến nay đã đạt tới phiên bản ASP.NET MVC 6

+2016 thì Microsoft đưa ra khái niệm .NET Core và ASP.NET Core MVC ra đời, tới nay đã có ASP.Net Core MVC 3

Trong đó gồm :

**-Model** : dùng C# hoặc VB.net để quản lý tốt dữ liệu. Nơi có thể truy cập view và controller. Nó là một cơ sở dữ liệu hoặc cũng có thể chỉ là một tập tin XML

**-View** tầng hiển thị, là 1 trang chứa cshtml. View được xem là một giao diện người dùng được sử dụng bởi khách hang truy cập

**-Controller:** là một lớp C# hoặc VB.net được kế thừa từ System.Web.Mvc.Controller. Controller được xem là thành phần tương tác của View và Model. Chứa các phương thức hành động chịu trách nhiệm xử lý URL. Có thể truy cập và truyền dữ liệu đến các khung nhìn khác

**4.1)Ưu điểm:**

-Nhẹ, tiết kiệm băng thông, việc này giúp cho các ứng dụng trên web tương tác gửi và nhận dữ liệu một cách liên tục để các web này hoạt động hiệu quả và ổn định

-Giúp kiểm tra dễ dàng, rà soát lỗi trước khi tới tay người tiêu dùng, đảm bảo chất lượng ổn định hơn

-Chức năng control: việc sử dụng mô hình MVC sẽ ưu việt trên nền tảng các ngôn ngữ hiện đại với nhiều hình thwusc khác nhau

-Chức năng Soc (Sêaration of Concern): cho phép phân tách rõ rang các thành phần Model, giao diện data, nghiệp vụ

-Tính kết hợp: thoải mái viết code trên website

-Đơn giản: là mô hình với kết cấu tương đối đơn giản, không cần sử dụng quá nhiều chuyên môn

**4.2)Nhược điểm:**

-MVC thường được dùng trong các dự án vừa và lớn, không thích hợp trong những dự án nhỏ. Dễ cồng kềnh và tốn thời gian trong quá trình phát triển cũng như trung chuyển dữ liệu

**5)Ứng dụng mô hình MVC vào lập trình**

-Mô hình MVC được dùng trong nhiều ngôn ngữ lập trình khác nhau, và được dùng phổ biến là ASP.Net MVC hay PHP MVC

-MVC đang phát triển rất tốt trên nhiều ứng dụng

\*\*Trước khi ASP.net MVC ra đời thì ASP.Net Web Forms là 1 framwork lập trình web được nhiều Developer sử dụng. Từ khi mà ASP.Net MVC ra đời thì cái ASP.Net Web From này bị bỏ quên và ưa chuộng mô hình MVC hơn

**6)ASP.Net được hiểu là gì?**

+Nó được coi là framework lập trình web được cũng cấp bởi ông lớn Microsoft và nó sử dụng C# để làm ngôn ngữ phát triển ASP.Net bao gồm 2 model đó là WEB FORMS và MVC

### **7)Tạo mới project ASP.NET MVC**

IDE là viết tắt của Intergrated Development Enviroment, tạm dịch là môi trường tích hợp và phát triển

+Sử dụng IDE các bản của Visual Studio

Ngoài các thư mục gồm Models, View, Controller còn có App\_Data, App\_Start, Content, font, Scripts,…

-App\_Data: là thư mục chứa các tệp dữ liệu liên quan đến ứng dụng như .mdf, LocalDB, and tệp XML ,…Điểm cần lưu ý ở thưu mục này là IIS sẽ không phục vụ các tệp có trong App\_Data

-App\_Start: là thư mục để chứa các tệp lớn, các lớp như BundleConfig, FilterConfig, RouteConfig, IdentityConfig, .. etc được lưu trữ trong thưu mục này

-Content: là thưu mục nội dung của MVC, ứng dụng này được dùng để lưu trữ tệp hình ảnh, tệp CSS và các icon files. Khi chúng tôi tạo một ứng dụng MVC 5, theo mặc định, bootstrap.css, Site.css, và các tệp bootstrap.min.css được bao gồm bởi studio trực quan như được hiển thị trong hình ảnh bên dưới

-Scripts: là thưu mục được sử dụng để chứa tất cả các file JavaScript cần thiết cho jquery và boostrap

-Fonts: dùng để chứa các font cần định dạng dùng trong MVC của boostrap

-Controller: thưu mục này chứa tất cả toàn bộ điều khiển. Thư mục Controller được kế thừa từ base Controller class

-Models: là thư mục sử dụng lưu trữ dữ liệu cũng như logic business để quản lý dữ liệu

-Views: là thư mục được sử dụng để chứa các tệp .cshtml là tệp mà có sự kết hợp của 2 ngôn ngữ html và c#

-Tất cả các thư mục nằm trong HomeController sẽ nằm trong thưu mục View

->Home

-Global.asax: là thư mục nằm trong ứng dụng MVC để viết mã mà để chạy trong ứng dụng

-Web.config: là thư mục nằm trong tệp hữu ích và quan trọng chứa các cấu hình cấp ứng dụng

## **II) ASP.NET VÀ MVC DESIGN PATTERN**

**1)Controller:**

+Là một lớp tập hợp các phương thức ở chế độ public. Các public methods của controller gọi là action methods hoặc simple methods. Nó sẽ yêu cầu các action methods trong mvc sẽ xử lý các yêu cầu mà http đến

+Controllers trong ASP.NET MVC là một nhóm các action tương tự nhau. Các controller sẽ xác định các bộ quy tắc như định tuyến, bộ nhớ và ủy quyền

**2)View trong ASP.Net MVC:**

- Trong MVC, view chứa logic để đại diện dữ liệu mô hình dưới dạng giao diện người dùng cuối có thể tương tác

- Thông thường, nó tạo giao diện người dùng với mô hình dữ liệu từ mô hình được bộ điều khiển cung cấp trong MVC. Vì vậy view trong ASP.Net MVC html được nhúng cú pháp Razor tạo nội dung html gửi tới máy khách

**2.1)Ưu điểm khi dùng View trong ASP.Net**

-View trong ASP.Net MVC cung cấp phân tách các code. Nghĩa là nó phân tách người dùng giao diện từ phần còn lại của ứng dụng

-ASP.NET MVC views sử dụng cú pháp Razor giúp dễ dàng chuyển đổi giữa mã HTML và C#

-Dễ dàng tái sử dụng bằng cách sử dụng bố cục và khung nhìn

**2.2)Razor là gì?**

Là một cú pháp kết hợp C# và HTML để sinh ra web động trong ASP.Net Core. Razor này được dùng trong Razor Pages, MVC và Blazor

**-Một số nguyên tắc dùng trong Razor:**

+Ngôn ngữ mặc định trong file cshtml là HTML: nếu không đánh dấu đặc biệt thì Razor (view engine) sẽ hiểu được nội dung bạn trình bày là ngôn ngữ HTML, và sẽ chuyển thẳng khối HTML này thành kết quả đầu ra

+Kí từ @ yêu cầu Razor chuyển từ HTML sang C#, nghĩa là nếu nhìn thấy @ thì razor sẽ hiểu là bạn đang muốn viết code C# và chuyển sang chế độ dịch C#

+Kết quả dịch mã cuối cùng của file Razor là mã HTML

\*HTML Helper:

-HTML Helper là một phương thức mở rộng của HTML. Lớp helper này được sử dụng để tạo nội dung html dưới dạng view

-Lớp HTML Helper tạo các phần tử html bằng cách sử dụng đối tượng lớp Modeltrong Razor View

-Nó liên kết đối tượng Model trong HTML để hiển thị giá trị thuộc tính Model trong HTML

**Cú pháp:**

@Html.TextBox(“firstname”, “ ”)

\*HTML Language :

<input id= ‘ Firstname’ name= ‘ fistname’ type= ‘text’ value = ‘Hung’/>

Một vài cú pháp dùng trong html helper:

@Html.Hidden(“id”)

@Html.Password(“Password”

@Html.Label(“Name”, “Name”)

@Html.TextArea(“Comments”, “”, 6, 3, null)

**3)ViewData- ViewBag**

-Trong ASP.Net MVC chúng có thể vượt qua dữ liệu chẳng hạn như sử dụng ViewBag, ViewData, TempData

-Bạn có thể sử dụng Session và Application Sate để sử dụng các biểu mẫu Web Form truyền thống

\*Sử dụng ViewData: được phân giải tại thời gian chạy, do đó nó không cung cấp lỗi thời gian biên dịch

-View Data trong ASP.Net chỉ truyền dữ liệu từ action method sang view

\*Dùng ViewBag trong ASP.Net MVC là một trong những cơ chế truyền dữ liệu cũng từ controller sang view. Lưu trữ và truy cập dữ liệu trong ViewBag dùng với kiểu chuỗi

**4)Sự giống và khác nhau của ViewBag và ViewData**

-Đều sử dụng ViewBag và ViewData để chuyển dữ liệu từ controller snag view

-Ở ViewData là một đối tượng từ điển còn ViewBag là một tài sản động

-ViewData và ViewBag sử dụng chuỗi làm khóa lưu trữ và truy xuất dữ liệu

-Ở ViewData yêu cầu sự chính xác về chính tả

### **III)Layout for View in ASP.Net**

**1)HTML Helper class**

-Cung cấp hỗ trợ để hiển thị biểu mẫu html, sử dụng html helper trong view để hiển thị nội dung.

-Là một phương thức trả về chuỗi

-Được chia thành 3 loại:

+Inline html helper

+Custom html helper

+Built-in html helper

-Riêng Built-in html helper chia thành 3 loại:

+Standard

+Strong Typed

+Template

**2)Standard Html Helper**: được dùng để hiển thị trong TextBox, DropDown, Radio buttons, Checkbox,…

\*Các phương thức mở rộng trong html helper: các lớp trợ giúp có một số phiên bản quá tải

-Standard HTML Helper:

+BeginForm- EndForm

**3)Cú pháp:**

@using (Html.BeginForm ())

{

@ Html.LabelFor (...)

@ Html.TextBoxFor (...)

. . . . .

. . . . .

@ \* --- Các phần tử của biểu mẫu có ở đây --- \* @

}

+HTML Helpers Label (gồm 2 phương thức là Label() và LabelFor())

**3.1)Html.Label()**

\*Cú pháp: MvcHtmlString Label(string expression, string labelText, object htmlAttributes)

**3.2)Html LabelFor()**

\*Cú pháp: MvcHtmlString LabelFor(<Expression<Func<TModel,TValue>> expression)

**3.3)Html TextBox()**

\*Cú pháp: MvcHtmlString TextBoxFor(<Expression<Func<TModel,TValue>> expression, object htmlAttributes)

**3.4)Html DropDownList()**

\*Cú pháp:

*MvcHtmlString DropDownList (stringname, IEnumerable < SelectLestItem> selectList, string optionLabel, object htmlAttributes)*

**3.5)Html DropDownListFor()**

\*Cú pháp:

*MvcHtmlString DropDownListFor ( < Expression< Func< TModel,TValue>> expression, IEnumerable< SelectLestItem> selectList, string optionLabel, object htmlAttributes)*

Ngoài ra còn có them một số cú pháp là: TextBoxFor(), Html.TextAreaFor(),

Html.DropDownListFor(), Html.CheckboxFor(), Html.RadioButtonFor(), Html.ListBoxFor(), Html.PasswordFor(), Html.HiddenFor(), Html.LabelFor(), ...

etc.

**3.6)Templated Html Helper:**

\*LayoutView

-Một số ứng dụng chứa các phần dùng chung trong suốt ứng dụng: logic, tiêu đề, thanh điều hướng bên trái, thanh bên phải hoặc phần footer

-Layout View cho phép xác định mẫu dùng chung

-Layout View kế thừa trong nhiều chế độ xem để cung cấp giao diện nhất quán

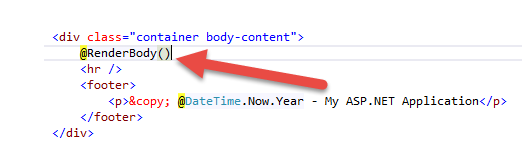
\*View\_Start.cshtml bao gồm thư mục chế độ xem mặc định. Nó thiết lập trang bố cục mặc định cho tất cả các chế độ xem trong thư mục và các thưu mục con bằng cách sử dụng thuộc tính

\*Rendering Methods:

-RenderBody()

+Trong \_Layout.cshtml sẽ có 1 thuộc tính là @RenderBody

+Hai trang About và Contact có hearder và footer giống nhau nhưng về phần nội dung sẽ khác nhau hoàn toàn, vì vậy LayoutPage cung cấp 1 thuộc tính là RenderBody().



Hinh2

-RenderSection: là một phương thức giúp xác định vị trí hiển thị nội dung trên trang web.

@RenderSection có hai giá trị trả về là false hoặc true. Thường thì nhiều Deverloper thường dùng kiểu trả về ‘false’ nó sẽ ít xảy ra lỗi hơn so với dùng ‘true’.

Ví dụ :

+ @RenderSection(‘footer’, required: false): nội dung hiển thị không bắt buộc

+@RenderSection(‘footer’, required: true): nội dung hiển thị bắt buộc, nếu không hiển thị sẽ xảy ra lỗi

**4)Nuget Package**

+Là một công cụ thiết yếu cho bất kì nền tảng hiện đại qua đó có thể tạo, chia sẻ và sử dụng mã hữu ích

+ ‘Package’ thường được chứa mã đã phiên dịch (dạng ddl) và với một số nội dung không cần thiết

+ Đối với .NET Microsoft hỗ trợ mã chia sẻ NuGet được lưu trữ và sử dụng đồng thời cung cấp từng vai trò

+Nuget Package được phát triển 2010 và được dùng trong Visual 2012

**5)Bundling and minification:**

-Buildling: là một file mà có thể được tham chiếu bởi duy nhất và được loaded bởi http

-Minification: là xử lý để bỏ đi các khoảng trống dư thừa, xuống dòng và comment trong code để giảm đi cỡ và cải thiện thời gian load

=> Các developer sử dụng file JS và CSS dẫn tới vấn đề performance cho web. Bởi các file này cần nhiều http request từ trình duyệt dẫn tới suy giảm về hiệu suất và thời gian load cho web

\*Các loại Bundling :

Trong MVC 5 gồm các lớp trong System.web.Optimization:

+ScriptBundle: chịu trách nhiệm về Javascript minification nhỏ hoặc nhiều file

+StyleBundle: chịu trách nhiệm rút gọn các file css của một hoặc nhiều style sheet files

+DynamicFolderBundle: đại diện cho một đối tượng Bundle ASP.Net tạo từ một thưu mục chứa tệp cùng loại

\*BundleConfig class và RegisterBundle

+BundleCollection

+StyleBundle

+ScriptBundle

\*Register bundle

+Global.asax

+Application\_Start event

### **IV)ENTITY fRAMWORK**

### **(1)ORM?**

ORM trong ASP.Net MVC là một phần thiết yếu và nó được hiểu là thiết kế kiến trúc.

Gồm Model-View-Controller (MVC).

-Model: được thiết kế để quản lý business logic

-View: chế độ xem mà người dùng có thể nhìn thấy

-Controller: quản lý sự tương tác giữa Model và View

=>Cần truy cập database từ lớp (Controller). Tại thời điểm đó, trình ánh xạ quan hệ đối tượng (ORM) sẽ xuất hiện

=>ORM là một ứng dụng hoặc hệ thống hổ trợ chuyển đổi dữ liệu trong hệ quản trị cơ sở dữ liệu và mô hình đối tượng cần thiết kế để sử dụng trong lập trình hướng đối tượng

\*ORM hỗ trợ trong:

+NHibernate

+Entity Framwork

+LinQ to SQL,...

**(2)ENTITY FRAMWORK**

-Là một chương trình ánh xạ qua lại giữa các object của một chương trình, bản gi, bảng cơ sở dữ liệu quan hệ. Và Entity Framwork được coi là một thư viện ORM. Hoặc hiểu đơn giản là một công cụ do Microsoft phát triển 2008 và trở thành công cụ làm việc với cơ sở dữ liệu

trong đó ORM chính là tập hợp công nghệ có thể làm việc với các hệ quản trị cơ sở dữ liệu

-Quá trình ánh xạ này thì ORM sẽ hỗ trợ tự động để xử lý sự chênh lệch giữa ngôn ngữ lập trình hướng đối tượng với cơ sở dữ liệu quan hệ

-Có thể dùng LinQ sau đó truy xuất và thao tác dữ liệu các đối tượng

**(3) Cách sử dụng:**

- Để sử dụng Entity Framework thì các nhà phát triển sử dụng một ngôn ngữ truy vấn dữ liệu thông qua ngôn ngữ C# gọi LINQ, mục đích là có thể đọc dữ liệu và thao tác trên dữ liệu.

+ Entity Framework : có thể tạo ra các câu lệnh truy vấn cần thiết để đọc hoặc ghi dữ liệu trong cơ sở dữ liệu và thực thi chúng cho bạn + Nếu cần truy vấn dữ liệu bạn có thể sử dụng LINQ và Enties 29: 25 - 31: Conceptual Model : Là 1 mô hình đối tượng

+ Có thể sử dụng Visual Designer để định nghĩa các mô hình khía niệm. Tự động sinh các Class

+ Dùng Code first định nghĩa các Class cần thiết theo mô hình khái niệm thì Entity framwork sẽ thực hiện chuyển đổi và nó sẽ tạo data cho người dùng luôn.

**(4) Kiến trúc :**

- Kiến trúc Entity Framework gồm 3 thành phần chính:

+ Object Service : làm trung gian để chuyển đổi các đối tượng object thành những entity

+ Entity Client : Nhiệm vụ chính của nó là chuyển đổi những truy vấn Entity SQL vào một truy vấn SQL

+ Data Providers : chuyển dữ liệu đi.

**(5) Code First - Model-first - Database-first là gì?**

- Code First viết các class Model bằng code C# Generate Database từ class Model không dùng đến Database và thông qua việc định nghĩa các class từ đó sinh ra Database

- Model-first thiết lập mô hình dựa trên EDMX.

- Database-first việc tạo mô hình thông qua các database có sẵn

\*Hiện nay mô hình Code First và Database First là phổ biến nhất.Vậy nó có gì khácnhau

|  |  |  |
| --- | --- | --- |
| Ưu và nhược điểm | Code First | Database First |
| Ưu điểm | Có lợi thế trong việc phát triển các dự án ở mức độ nhỏ | Dễ xây dựng mô hình dữ liệu, hướng đến các dự án lớn |
|  | Rất phổ biến (vì các lập trình viên thường không thích thiết kế DB, nhưng thích thiết kế class) | Entity Framework sẽ tạo ra các Entity class cho bạn |
|  | Kiểm soát hoàn toàn code model, thêm xóa sửa thuộc tính vô cùng dễ dàng | DB có thể được develop riêng  Dùng được DB có sẵn |
| Nhược điểm | Các thay đổi cấu trúc trực tiếp trên DB sẽ mất | Không thể thay đổi code đã được Generate (nó sẽ mất trong lần chỉnh sửa cấu trúc DB tiếp theo) |
|  | Khó kiểm soát những column sẽ tạo trên Db | Khó khăn khi muốn thêm các DataAttribute và DisplayAttribute cho các class model |
|  | Hơi khó khi kết hợp với Db có sẵn | Mỗi lần thay đổi cấu trúc DB, bạn sẽ phải update lại EDMX và tạo lại các class Model để phản ánh sự thay đổi đó |
|  | Chỉ sử dụng được trong những dự án nhỏ | Không thuận tiện trong việc sử dụng trong các dự án nhỏ |

**(6)Các toán tử trong *Entity Framework ?***

+ Filtering Operators Lọc dữ liệu, có 2 cách: Where và OfType (chọn ra giá trị thuộc 1 kiểu nào đó)

+ Sorting Operators Sắp xếp dữ liệu, có các phương thức: OrderBy (tăng dần), OrderByDescending (giảm dần), ThenBy (tăng dần xét thêm yếu tố thứ 2) và ThenByDescending (giảm dần xét thêm yếu tố thứ 2)

+ Grouping Operators Lập nhóm dữ liệu, dùng GroupBy

+ Conversions Chuyển đổi kiểu của đối tượng đầu vào

+Aggregation Thống kê trên nhóm dữ liệu: Max (tìm giá trị lớn nhất), Min (tìm giá trị nhỏ nhất), Sum (tính tổng), Average (tính trung bình cộng), Count (đếm số phần tử),…

+Set Operations Các phép toán trên tập hợp: Distinct (không trùng), Except (ngoại trừ), intersect (phép giao), union (phép hội)

+Element Operators Lấy ra 1 phần tử nào đó từ collection: ElementAt, ElementAtOrDefault, First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault, ngoại trừ DefaultIfEmpty

+Take và Skip các phần tử trong 1 chuỗi Take: lấy ra số phần tử theo ý muốn và bỏ qua các phần tử còn lại Skip: bỏ qua số phần tử theo ý muốn và lấy ra các phần tử còn lại Kết hợp Take và Skip để phân trang

+TakeWhile Operator TakeWhile lấy ra số phần tử khi điều kiện xác định vẫn đúng. Một khi điều kiện sai, TakeWhile sẽ dừng mà không cần kiểm các phần tử còn lại dù điều kiện có thể đúng cho các phần tử còn lại.

**(7) Lambda Expressions là gì?**

-Lambda Expressions Dùng để mô tả dữ liệu khi dùng phương thức LINQ

**(8)Method Syntax có công dụng gì?**

-Method Syntax Ngoài dùng query, có thể dùng các phương thức như Where, Select, SortBy, GroupBy, Join,… Ví dụ giữa dùng query và dùng phương thức

## **V)LINQ**

**(1) Khái niệm về LINQ?**

- LINQ (Language Integrated Query) là ngôn ngữ truy vấn tích hợp trong C# và [VB.NET](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbkZnZTBJWXRlMFlneUxnNTJrdHVQSXZMYS1vQXxBQ3Jtc0tudzhNYVMxUmFBV09XWXkySGdXLWpYWGdwbXZPeks0VWJfdEI2LVkzX211SUlkOUZIdEZ2YWFXaFBRZ2U3dFJoOEFqcHJ0aW56eDdEMnoxMXpDQk94bWFOaml5RjVCMU54azdGcGJfU3ZaT0F6djFTZw&q=http%3A%2F%2Fvb.net%2F&stzid=UgxKuh9QdbJUhXEM5I14AaABAg" \t "_blank) thường được dùng để lưu trữ và truy suất dữ liệu từ nhiều nguồn. Dữ liệu ở đây có thể được bắt nguồn từ bộ nhớ, XML, cơ sở dữ liệu (Mysql, Sql Server...)... hay bất cứ nơi lưu trữ nào khác. Bất cứ dữ liệu đó bắt nguồn từ đâu, chúng ta phải luôn luôn thực hiện việc truy vấn dữ liệu đến nó. Với mỗi nguồn dữ liệu khác nhau sẽ có những cách khác nhau để thực hiện truy vấn. LINQ ra đời nhằm thực hiện hợp nhất cách truy vấn thành 1 để có thể dùng chung cho các nguồn dữ liệu khác nhau. LINQ.

-Các loại LINQ:

+ LINQ to Objects

+ LINQ to SQL

+ LINQ to Entities

+ LINQ to DataSets

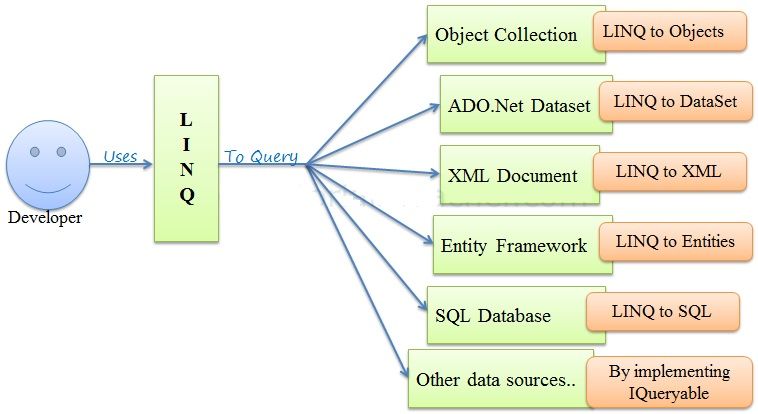
+ LINQ to XML

**(2) Cấu trúc của LINQ**

- Cấu trúc của LINQ trong .NET LINQ có 3 lớp cấu trúc:

+Tầng trên cùng là lớp chứa các ngôn ngữ mở rộng như C\*, VB, và một số ngôn ngữ .NET khác.

+Tầng dưới cùng chứa các nguồn dữ liệu đối tượng thực thi các interface với kiểu dữ liệu generic Ienumberable<T> hoặc IQueryable<T>



Hinh3

**(3) Công dụng của LINQ**

- LINQ cung cấp chức năng làm nỗi bật cú pháp giúp ích trong quá trình tìm kiếm sửa chữa lỗi. LINQ cung cấp chức năng nhắc lệnh thông minh(IntelliSense) giúp việc truy vấn được chính xác và nhanh chóng hơn.

- Sử dụng LINQ giúp viết code nhanh hơn rất nhiều, giúp giảm thời gian đáng kể quá trình phát triển Sản phẩm.

- LINQ giúp sửa lỗi dễ dàng hơn do tích hợp ngôn ngữ C# Với tính năng phân cấp bậc giúp xem mối quan hệ giữa hai bảng dễ hơn, đồng thời giúp việc truy vấn kết hợp nhiều bảng trong thời gian ngắn hơn.

- LINQ chỉ cho phép sử dụng duy nhất một cú pháp LINQ trong khi truy vấn từ nhiều nguồn bởi nền tảng thống nhất của nó.

- LINQ có thể mở rộng bằng các sử dụng kiến thức về LINQ để truy vấn các nguồn dữ liệu mới.

- LINQ cung cấp cơ sở kết hợp nhiều nguồn dữ liệu trong một truy vấn duy nhất cũng như chia các vấn đề phức tạp thành một tập hợp các truy vấn ngắn dễ gỡ lỗi.

- LINQ cung cấp khả năng chuyển đổi dễ dàng để chuyển đổi kiểu dữ liệu này sang kiểu dữ liệu khác như chuyển đổi dữ liệu SQL sang dữ liệu XML.

**(4) Các từ khóa được sử dụng trong LINQ:**

- From: lấy ra từ nguồn nào

- Where: lọc dữ liệu

- Orderby: sắp xếp

- Groupby: nhóm tập dữ liệu có đặc điểm chung

- Select: chọn dữ liệu

**(5) Truy vấn LINQ**

- Truy vấn LINQ Query là biểu thức để nhận dữ liệu từ các nguồn dữ liệu thông qua các câu truy vấn. Với mỗi một loại dữ liệu phải có cách truy vấn khác nhau khiến người lập trình phải học nhiều ngôn ngữ truy ván khác nhau cho mỗi một loại dữ liệu LINQ dựa trên nguyên tắc tạo ra mô hình làm việc thống nhất nhưng có thể thao tác được trên nhiều loại dữ liệu khác nhau.

-Trong LINQ, ta luôn được làm việc với các objects. 3 bước để thực hiện một truy vấn:

Bước 1. Xác định loại dữ liệu

Bước 2. Viết câu truy vấn

Bước 3. Thực thi câu truy vấn đó

\*Các kiểu truy vấn LINQ:

-Truy vấn linq bằng Querty:

*IEnumerable<Tên Dữ Liệu> Tên biến= Têndữliệus.Where(Tên biến Lambda Expressions=> truy vấn).OrderBy(Tên biến Lambda Expressions=> n);*

*Select tên biến;*

-Truy vấn linq bằng Method

**(6) Toán tử trong LINQ**

- Toán tử Filtering (lọc dữ liệu) - Thực hiện việc lọc dữ liệu với hai từ khóa Where và OfType với Where lọc dữ liệu trên một hàm thuộc tính và OfType lọc dữ liệu trên một kiểu cụ thể.

- Toán tử sắp xếp(Sorting) - Việc sắp xếp dựa trên các toán tử: + OrderBy: sắp xếp dữ liệu tăng dần.

+ OrderByDescending: sắp xếp dữ liệu giảm dần.

+ ThenBy: thực hiện sắp xếp thứ cấp theo thứ tự tăng dần.

+ ThenByDescending: thực hiện sắp xếp thứ cấp theo thứ tự giảmdần.

- Tìm hiểu phép toán tạo group: dùng GroupBy để Grouping các dữ liệu.

- Thao tác chuyển đổi kiểu dữ liệu.

- Việc chuyển đổi thông qua các phép toán như AsEnumerable, AsQueryable, Cast, ToList, ToLookup....

- Aggregation:

+ Aggregate thực hiện thồng kê.

+ Average tính giá trị trung bình.

+ Count đếm số lượng

+ LonCount đếm số lượng với mức độ lớn.

+ Max tính giá trị lớn nhất.

+ Min tính giá trị nhỏ nhất.

+ Sum tính tổng giá trị.

- toán trên tập hợp với các toán tử Distinct, Except, Intersect, Union.

- Phép toán Element với các toán tử ElementAt, ElementAtOrDefault, First, Last...

- Take trả về một số phần tử nhất định trong một dãy và bỏ qua phần còn lại. - Skip sẽ ngược lại với Take, nó bỏ qua một số phần tử nhất định trong một dãy và trả về phần còn lại.

**(7) Tìm hiểu Lambda Expression**

- Thuật ngữ 'biểu thức Lambda' bắt nguồn từ tên của nó từ phép tính 'lambda', đến lượt nó là một ký hiệu toán học được áp dụng để xác định các hàm. Các biểu thức Lambda như một phần thực thi của phương trình LINQ dịch logic theo cách tại thời điểm chạy để nó có thể chuyển đến nguồn dữ liệu một cách thuận tiện. Tuy nhiên, biểu thức lambda không chỉ giới hạn để tìm ứng dụng trong LINQ.

(Tên Biến Lambda) ⇒ Tên Biến trong database cần truy vấn

\* Lưu ý: Không thể so sánh LINQ với Entity bởi LINQ là ngôn ngữ dùng trong việc khai thác dữ liệu, còn Entity là data model được ánh xạ từ dữ liệu Database thành các đối tượng.

## **VI)STORING AND TRACKING**

### **(1)IMPLICIT OBJECT**

Hoạt động của 1 ứng dụng web được xây dựng bởi ASP.NET, các thông tin có liên quan đến quá trình hoạt động của ứng dụng như:

- Thông tin yêu cầu từ phía người dùng (Client machine)

- Các đối tương lưu trữ thông tin về quá trình làm việc trong hệ thống (Session)

- Các biến toàn cục lưu trữ trạng thái của ứng dụng, ... (application object)

Tất cả đều được lưu trữ trong những đối tượng quy ước và xây dựng sẵn trên nền

tảng của .NET Framework, người lập trình hoàn toàn có thể khai thác dựa trên các

thuộc tính, phương thức của những đối tượng này, chúng bao gồm:

+ Request Object

+ Response Object

+ Server Object

### **(2)REQUEST OBJECT – HTTPREQUESTBASE CLASS**

**-Request:** chính là sự tương tác giữa trình duyệt chạy trên máy tính truy cập với máy chủ web thông qua 1 trang web nào đó thuộc chương trình, URL được gửi đi từ trình duyệt sẽ đại diện cho các yêu cầu gửi về máy chủ

**-HttpRequest** : là 1 lớp được cung cấp sẵn trong thư viện (namespace) System.web

thuộc .NET Framework, các thuộc tính của lớp này chủ yếu phục vụ cho việc đọc

thông tin từ phía Client.

**+ Một số các thuộc tính tiêu biểu**

* **ApplicationPath**: trả về “đường dẫn ảo” (Virtual path) của ứng dụng web
* **Browser**: Được dùng để truy xuất thông tin về trình duyệt mà người dung đang sử dụng để truy cập ứng dụng. VD: Tên trình duyệt, phiên bản, nền tảng của trình duyệt (Webkit, Mozilla, ...):
* **ContentLength**: Trả về giá trị số mô tả cho kích thước của dữ liệu được gửi về từ
* **FilePath**: Tương tự như ApplicationPath, tuy nhiên bao gồm cả tên trang aspx cần truy cập
* **QueryString**: Thuộc tính này trả về giá trị tượng trưng cho tham số của Request (Request parameter) với các cặp giá trị theo kểu “name=value”, trong đó ngăn cách giữa các cặp tham số là dấu “&”
* **RequestType**: Đây là thuộc tính mô tả cho phương thức truyền dữ liệu của form dựa trên HTTP: “get” hoặc “post” .

**+ Phương Thức**

* **SaveAs():** Lưu lại nội dung của Request và ghi thành tập tin trên đĩa
* **MapPath():** Trả về chuỗi giá trị mô tả cho “đường dẫn vật lý” của trang aspx ở trên đĩa dựa vào “đường dẫn ảo” thông qua request

## **(3) RESPONSE OBJECT**

**Response** cũng là đối tượng dùng để tương tác giữa máy chủ web với trình duyệt tại máy truy cập, tuy nhiên nhiệm vụ của Response là chuyển thông tin từ máy chủ web xuống cho client (Khác với Request là chuyển thông tin từ Client về cho máy chủ)

ASP.NET cung cấp các đối tượng Resopnse thuộc lớp HttpResponse thuộc .NET Framework với các thuộc tính tiêu biểu như:

* **BufferOutput:** Quy định việc có sử dụng bộ đệm để chuyển thông tin từ máy chủ web xuống client hay không
* **CharSet:** Chỉ ra định dạng đối với dữ liệu cần thể hiện trên trình duyệt

(Unicode, ISO-8859-1, ...)

* ContentType: Mô tả loại dữ liệu sử dụng đối với nội dung gửi xuống

Client

* **Cookies:** Thuộc tính này cho phép tương tác với cookies tại máy truy

cập (Lưu thông tin vào cookies của client)

* **IsClientConnected:** Trả về trạng thái kết nối của Client trong quá trình

đang chuyển dữ liệu từ máy chủ web xuống

**+ Phương thức:**

* **Clear():** Xoá toàn bộ nội dung trong bộ đệm
* **Close():** Đóng kết nối với client
* **End():** Gửi toàn bộ dữ liệu hiện có trong buffer xuống client
* **Redirect():** Chuyển thi hành tới 1 trang khác thông qua URL được chỉ định
* **Write():** Ghi thông tin xuống Client

**(4)SERVER OBJECT**

**Server** object là đối tượng của lớp HttpServerUtilityBase thuộc .NET Framework. Server object được trang bị sẵn các phương thức phục vụ cho việc kiểm soát, điều khiển giữa các trang trong ứng dụng web, phân tích HTML gửi về, ...

**+ Thuộc tính:**

* **MachineName:** thuộc tính mô tả thông tin của máy chủ web hiện hành
* **ScriptTimeout:** Dùng để truy xuất thời gian chờ tối đa khi chạy script (Time out)

**+ Phương Thức:**

* **Execute():** Phương thức cho phép gọi thi hành mã lệnh phíaServer của 1 trang web.
* **HtmlEncode():** Dùng để chuyển đổi chuỗi yêu cầu thành mãHTML
* **MapPath():** Trả về “đường dẫn vật lý” (Đường dẫn lưu trữ thực tế) củatrang ASP ở trên đĩa
* **UrlEncode():** URL chỉ có thể được gửi qua Internet bằng cách sử dụng bộ ký tự ASCII. Mã hóa URL thay thế các ký tự ASCII không an toàn bằng "%" theo sau là hai chữ số thập lục phân.

**Application object** là đối tượng phục vụ cho việc lưu trữ dữ liệu xuyên suốt ứng dụng web. Các đối tượng dữ liệu được lưu trữ ở dạng này có thể tồn tại trong bộ nhớ trên máy chủ từ khi ứng dụng web bắt đầu thi hành cho đến khi ngừng hoạt động. Có thể nói thời gian tồn tại của Application object được xem là dài nhất, tương ứng với vòng đời của 1 ứng dụng web

Application object là những thể hiện của lớp **HttpApplicationState** class được định nghĩa sẵn trong .NET Framework tại thư viện System.Web.

**+ Tính Chất:**

* **AllKeys:** cho phép truy xuất danh sách “tên” của các đốitượng Application thuộc ứng dụng hiện hành
* **Count:** Trả về giá trị số tương ứng với số lượng applicationobject đã được tạo trong ứng dụng
* **Item:** truy xuất đến các Application object dựa trên chỉ số (Tương tự như mảng hay Collection)

**+Phương Pháp**

* **Add()** : Thêm mới 1 Application object vào ứng dụng hiện hành
* **Clear():** Xoá bỏ toàn bộ các đối tượng có trong Application

object

* **Remove():** Xoá 1 application object thông qua tên gọi
* **RemoveAll():** Tương tự như Clear()
* **RemoveAt():** Xoá 1 application object dựa vào chỉ số (Vị trí lưu trữ

trong danh sách)

**(5)COOKIE**

**Cookies** là những thông tin ở dạng văn bản được gửi từmáy chủ web xuống và lưu trữ trên máy tính của khách truy cập.

**Cookies** rất hữu ích khi ứng dụng web cần lưu vết người dùng (Tracking) khi làm việc trên ứng dụng trong một thời gian dài (Từng ngày này, qua ngày khác)

++ *Việc lạm dụng Cookies lại có nguy cơ xâm phạm đến quyền riêng tư của người dùng Internet và điều này có thể dẫn đến “phạm luật”*

**5.1)Create Cookies**

*+ Sử dụng HttpCookies class*

*+Gán dữ liệu trực tiếp*

*+Chứa thông tin nhiều thành phần*

**5.2)Reading Cookies**

Để đọc thông tin từ Cookies của máy truy cập, ta sử dụng đối tượng Request

*+ Đọc cookies nhiều thành phần*

**5.3)Storing Cookies**

Thông tin trong cookies có thời hạn tuỳ thuộc vào thuộc tính Expires và được lưu trữ trên đĩa cứng của máy truy cập (Máy tính của người dùng) tại thư mục cookies, dưới dạng các file Text (\*.txt).

**5.4)SESSION OBJIECT**

Cũng giống như Application object, Session object có thể lưu bất cứ dữ liệu thuộc loại nào từ Primitive data type cho đến reference type, tuy nhiên, thời gian duy trì các đối tượng thuộc dạng này tối đa chỉ tương ứng với thời gian làm việc của người dùng trên hệ thống (Tính từ thời điểm truy cập cho đến khi tắt trình duyệt). Session object là nhữngthể hiện của **HttpSessionStateBase** class

**+Tính Chất :**

* **Count**: Xác định số lượng Session Object đang sử dụng trong ứng dụng
* **IsNewSession:** Trả về giá trị logic tượng trưng cho trạng thái của Session object hiện hành. Nếu Session mới được tạo thì trả về giá trị true, ngược lại là False
* **Keys:** Danh sách tên của các Session Object hiện có trong hệ thống
* **SessionID:** Trả về chuỗi giá trị được xem là “định danh” (Identify) của Session object hiện hành

**+Phương Pháp:**

* **Add:** Thêm, tạo mới 1 Session obejct để lưu trữ thông tin theo phiên làm việc của người dùng trong ứng dụng
* **Clear:** Xoá toàn bộ các Session object hiện có
* **Remove:** Xoá 1 session object dựa vào tên của nó
* **RemoveAt:** Xoá 1 Session object bằng cách chỉ ra vị trí (Chỉ số)
* **RemoveAll:** Tương tự như Clear, dùng để xoá tất cả các Session object hiện có trong hệ

thống

* **Abandon:** Phương thức này dùng để thực hiện kết thúc phiên làm việc của người dung một cách chủ động. Khi phương thức này được gọi, các sự kiện **Session\_End**, Sessin\_start, cùng các thông số như SessionID sẽ tự động được “làm mới” lại (Refresh)

***Có hai sự kiện quan trọng liên quan đến Session object***

**- Session\_Start:** sự kiện này xảy ra khi session bắt đầu được khởi tạo vào thời điểm người dùng truy xuất và hệ thống

**- Session\_End:** Đây là sự kiện xảy ra khi người dùng thoát khỏi hệ thống hoặc phiên làm việc của người dùng rơi vào trạng thái Session Time-out

*Cả 2 sự kiện trên đều được lưu trữ trong tập tin quản lý biến môi trường của ASP.NET và lập trình viên có thể viết mã lệnh thi hành cho các sự kiện này trong tập tin có tên Global.asax*

**\*\*GLOBAL.asax File**

Là Module dùng để lưu trữ và quản lý mã nguồn cho việc xử lý các sự kiện có liên quan đến Application Object và Session object như

* **Application\_Start**
* **Application\_End**
* **Session\_Start**
* **Session\_End**

**PHẦN II- KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG**

### **I-PHÂN TÍCH YÊU CẦU ĐỀ**

**Phần 1: TÊN ĐỀ TÀI**

“Xây dựng Website bán đồ online: Fashion & Beauty”

**1.CHỨC NĂNG**

Đây là một website bán và giới thiệu sản phẩm của cửa hàng cho người tiêu dùng giá cả thực của sản phẩm. Có các chức năng sau:

-Hiển thị danh sách các mặt hàng

-Cho phép tìm kiếm thông tin một mặt hàng nào đó hoặc một nhóm hàng

-Khách hàng có thể tự đăng ký thông tin của mình để trở thành khách hàng của công ty

-Cho phép người dùng tự cập nhật, sửa, xóa, thêm vào giỏ hàng mặt hàng mà mình yêu thích

-Cập nhật các nguồn hàng có từ đâu

-Sản phẩm đượ chia ra:

+Dành cho trẻ em(kids)

+Dành cho phụ nữ(women)

+Dành cho nam(men)

## **2.YÊU CẦU ĐẶT RA**

**a.Thiết bị và phần mềm**

-Máy tính có thể thiết kế được web

-Hệ quẩn trị cơ sở dữ liệu MySQL

-Phần mềm thiết kế web: Visual Studio 2017

**b.Các thành phần của trang web**

Hệ thống có 2 phần:

\*Thứ nhất :Dành cho người quản trị(Admin): Người làm chủ ứng dụng có quyền kiểm soát mọi hoạt động của hệ thống. Admin sẽ được cấp user và password để có thể sử dụng hệ thống của mình

-Chức năng cập nhật, sửa, xóa các mặt hàng, loại hàng, nhà sản xuất,… đòi hỏi phải cung cấp chính xác

-Tiếp nhận và thống kê các đơn hàng đặt

-Thống kê lại số lượng

\*Thứ hai: Dành cho khách hàng

-Khách hàng có thể mua sắm và thoải mái lựa chọn các sản phẩm có mặt trong website, họ có thể tìm kiếm và đặt mua. Đòi hỏi phải có các chức năng sau:

+Hiển thị danh sách các sản phẩm được phân loại dành cho các lứa tuổi và giới tính

+Khách hàng có thể xem các sản phẩm sale, các sản phẩm mới về

+Khách hàng chọn mặt hàng mình muốn mua và tiến hành đặt hàng và có thể xem lại thông tin đơn đặt hàng

## **II- PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

### **1)KHẢO SÁT HIỆN TRẠNG VÀ XÂY DỰNG HỆ THỐNG**

1.1)Khảo sát hiện trạng hệ thống

Cửa hàng thời trang Fashion & Beauty là một cửa hàng chuyên về các sản phẩm thời trang hiện nay. Đáp ứng cho mọi nhu cầu từ trẻ em, thời trang nữ và bao gồm thời trang nam. Và chất lượng và giá thành đảm bảo rất cao và không bao giờ làm khách hàng thất vọng. Cơ cấu của cửa hàng bao gồm:Quản lý, giao hàng, Marketing và kế toán.

Với phương châm “vừa lòng khách đến, vui long khách đi”, công ty của chúng tôi không ngừng nổ lực để đme đến những trải nghiệm hay và độc lạ cho khách hàng.

Với hai hình thức chọn hàng trên website và đi đến cửa hàng để lựa chọn. Thì những khách hàng thường mua trwucj tuyến nhiều hơn so với khách hàng mua đồ trực tiếp. Bên cạnh sự uy tín của công ty thì mau đồ trên website giúp ích và tiện lợi hơn nhiều. Bạn chỉ cần chọn sản phẩm mà bạn yêu thích sau đó đặt hàng và chọn hình thức thanh toán. Sau đó bên quản lí của cửa hàng sẽ đều phối người đi giao hàng tới địa chỉ mà khách hàng đã đặt

Từ đó doanh thu của cửa hàng tang đều theo hàng tháng nhờ hình thức bán hàng trên website. Chất lượng và thương hiệu của cửa hàng được nhiều người biết đến và khách quen cũ ngày càng tăng lên đáng kể.

**II)MỤC TIÊU XÂY DỰNG WEBSITE**

**1.Đối tượng của website**

Website được xây dựng nhầm phục vụ hai đối tượng là admin và khách hàng với các chức năng sau:

\*Admin:

-Đăng nhập hệ thống

-Xem, cập nhật, xóa thông tin sản phẩm

-Quản lý đơn hàng

-Xem và trả lời các đóng gớp từ khách hàng

-Xem xóa thông tin của khách hàng

-Cập nhật tin tức

\*Khách hàng:

+Guest:Khách ghé thăm

-Xem được thông tin của sản phẩm

-Đăng ký thành viên

+User:

- Đã có tài khoản đăng nhập:

-Có quyền đăng nhập, đăng xuất, đổi mật khẩu

-Đăt mua sản phẩm

-Được thông báo các sản phẩm mới và tình trạng của đơn hàng nếu khách hàng có đơn mua

**2.Đặc điểm**

Xây dựng một hệ thống bán hàng trực tuyến đơn giản, thân thiện, dễ dàng sử dụng

-Website được thiết kế :

+Giao diện thân thiện xem được các sản phẩm mọi lúc mọi nơi khi có kết nối giữa thiết bị và internet

+Khách hàng có thể tìm thấy thông tin sản phẩm dễ dàng

Khách hàng có đóng góp hoặc phản hồi đều được chúng tôi ghi nhận để hoàn thiện hơn dù tốt hay xấu

Các module:

-Module sản phẩm: hiển thị thông tin và các sản phẩm về các hình ảnh, nội dung, giá cả,…

-Moduel giỏ hàng: Khi đã tham khảo thông tin của snar phẩm thì khách hàng có thể chọn đặt mua và sau đó đi đến giỏ hàng tiến thanh toán là đã mua. Cần ghi đầy đủ thông tin để cung cấp cho bên giao hàng được giao chính xác nhất

-Module đăng kí và đăng nhập: khách hàng đăng xong là được tiến hành đặt mua và được hương một số voucher từ cửa hàng

-Còn các moduel khác …

**PHẦN III: PHÂN TÍCH HỆ THỐNG**

**I)Các thông tin đầu ra của hệ thống**

Thông tin đầu vào:

-Thông tin khách hàng

-Thông tin sản phẩm

-Thông tin về thương hiệu,…

Thông tin đầu ra

-Chi tiết sản phẩm

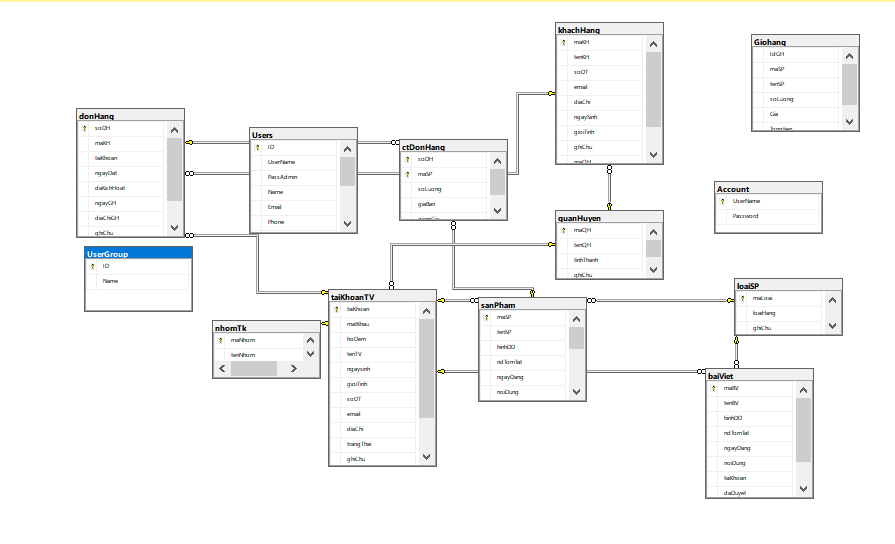
-Hóa đơn

-Các phản hồi

**1)MỘT SỐ HÌNH ẢNH GIAO DIỆN CHƯƠNG TRÌNH**

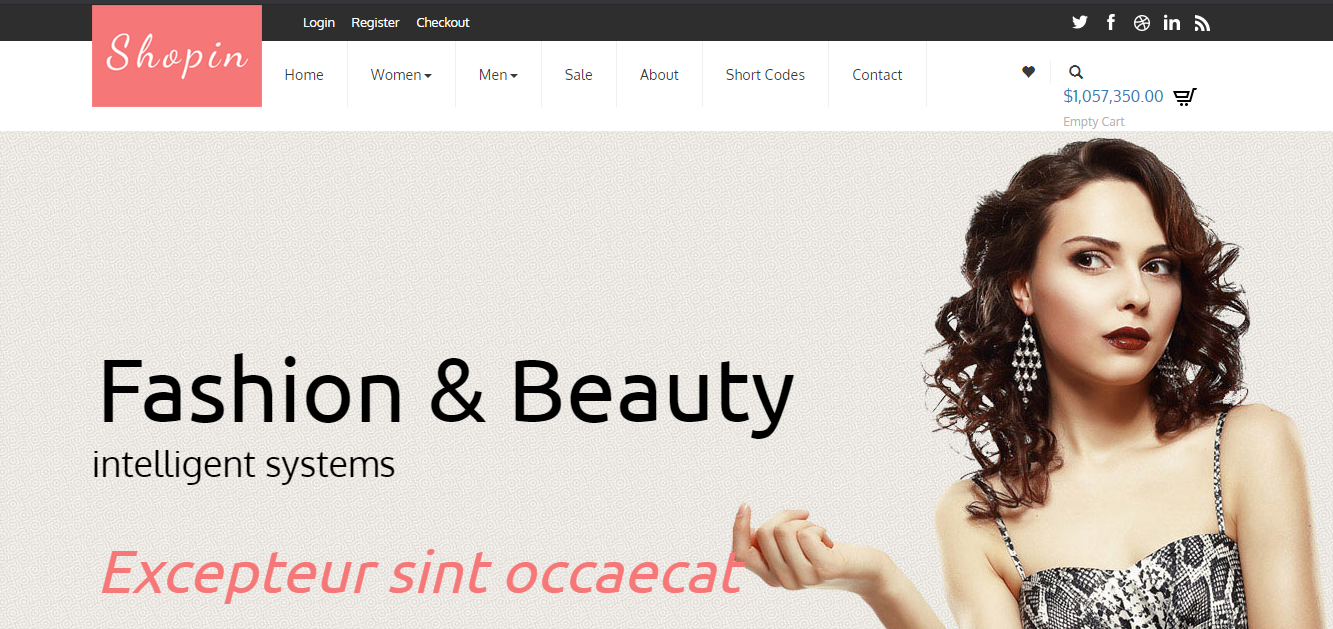
**A)Một số hình ảnh giao diện website**

\*Phần Index->Home

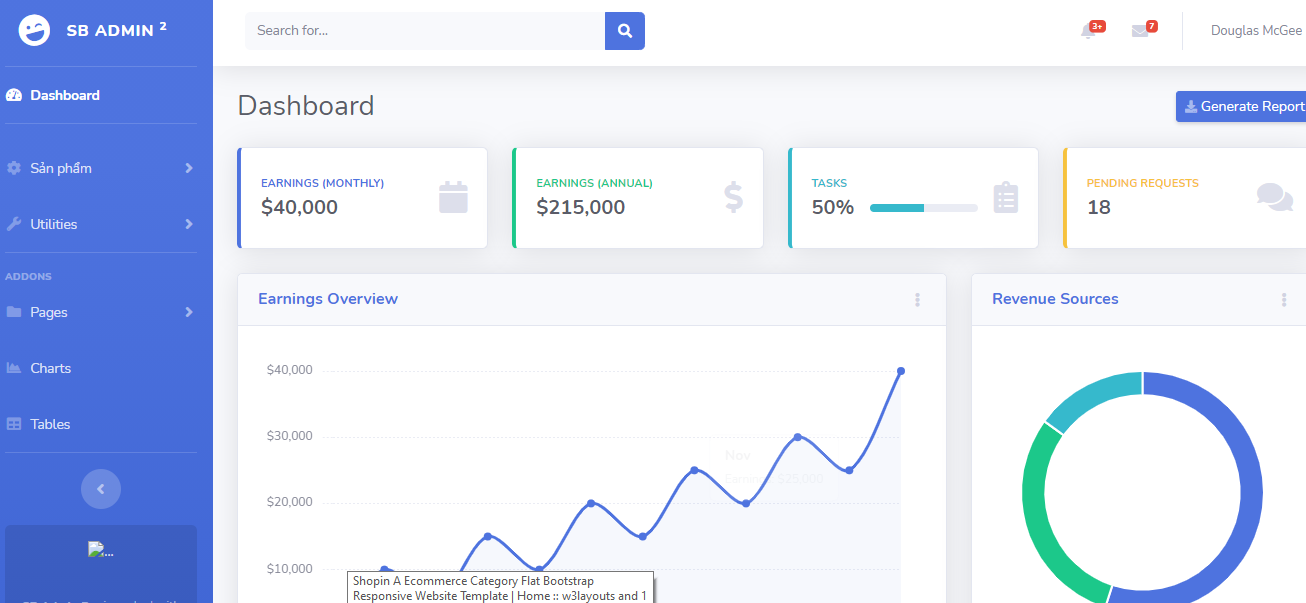


*Hinh4*

\*Giao diện trang cho phép khác hàng Chọn hàng

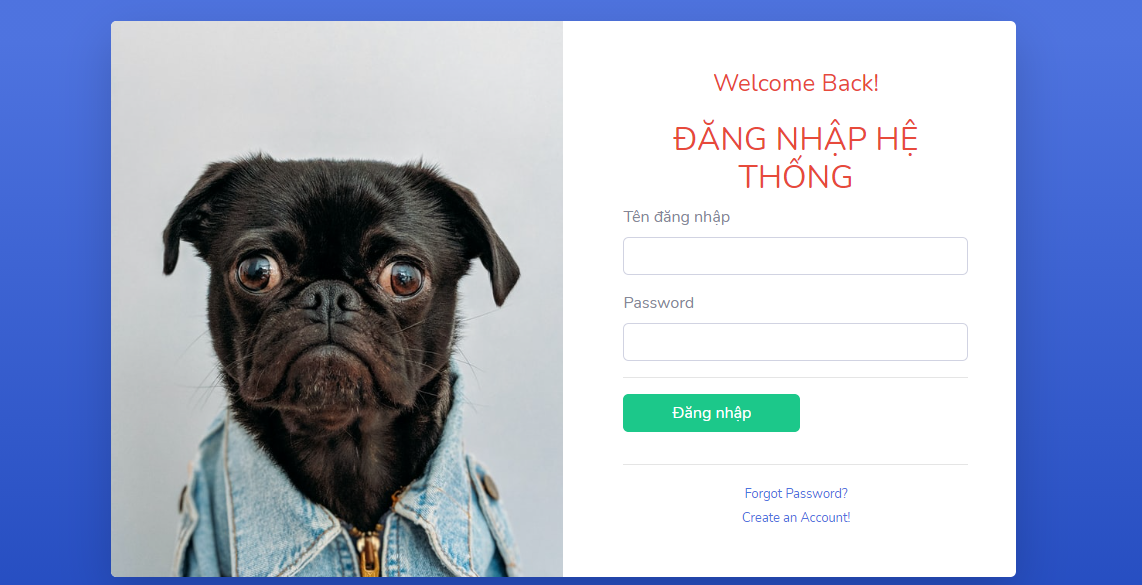


*Hinh5*



*Hinh6*

\*Giao diện phần login của admin



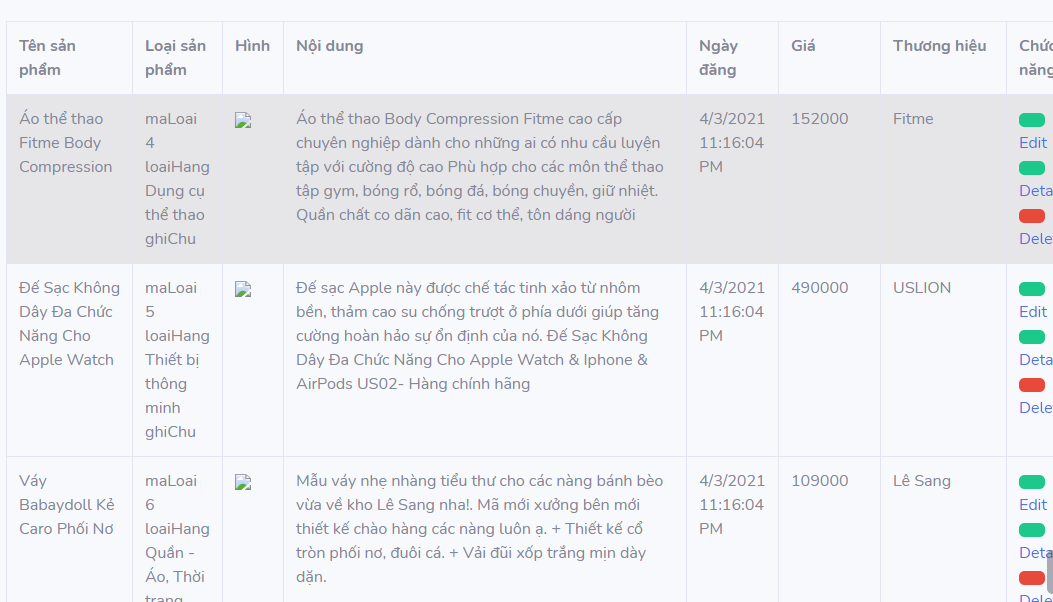
*Hinh7*

\*Phần thêm sản phẩm của admin



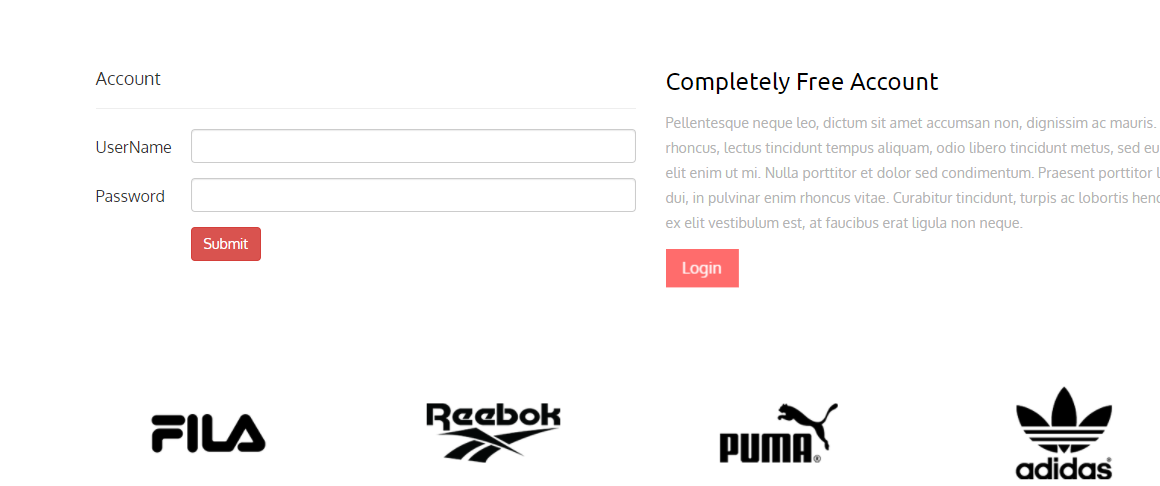
*Hinh8*

\*Chi tiết tất cả các sản phẩm trong admin



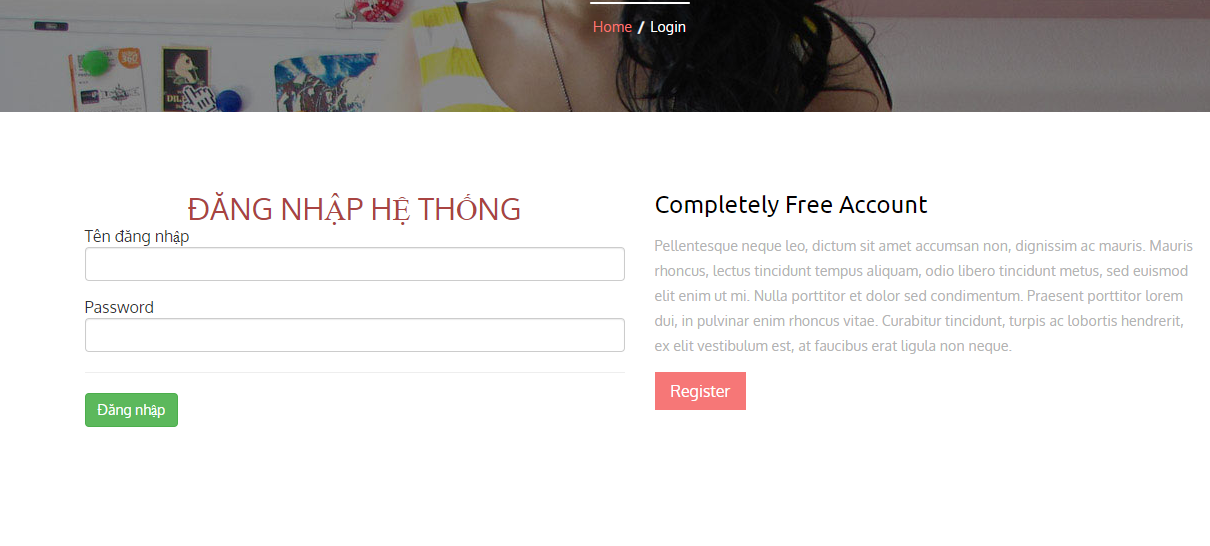
*Hinh9*

\*Giao diện phần đăng kí của User



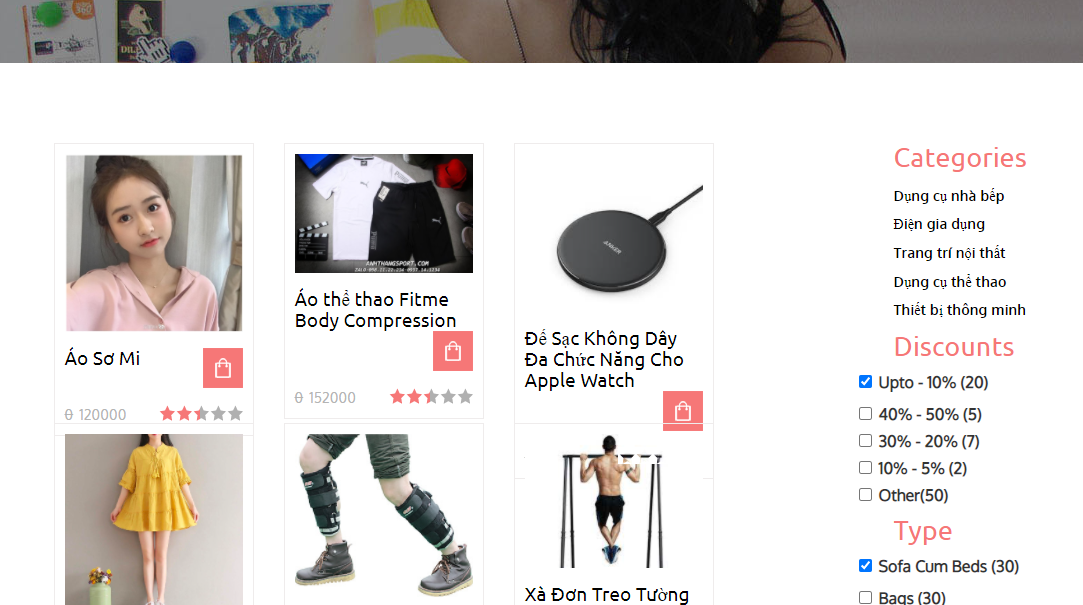
*Hinh10*

\*Giao diện phần đăng nhập

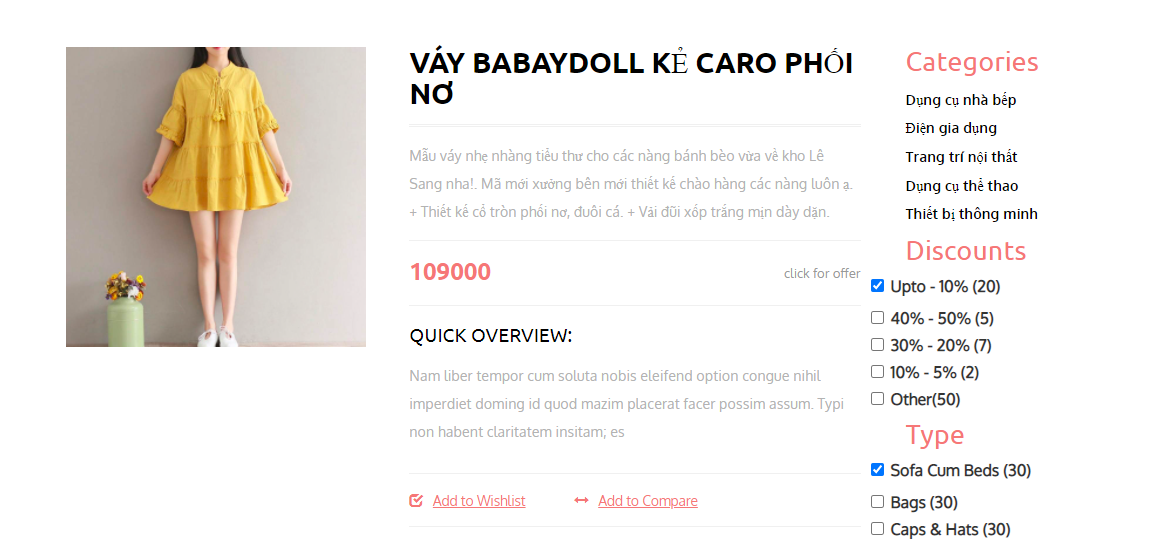


*Hinh11*

\*Trang sản phẩm



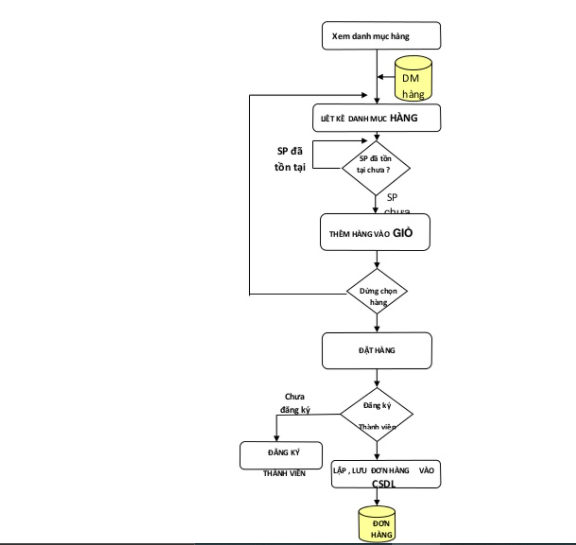
*Hinh12*



*Hinh13*

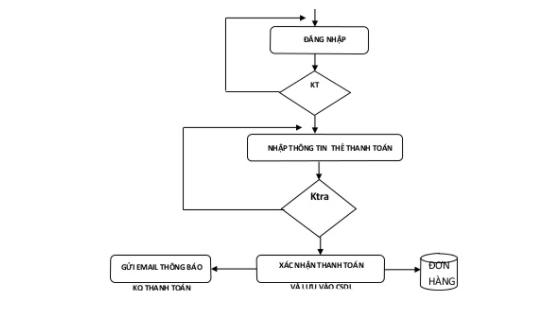
a)Sơ đồ CHỌN HÀNG VÀ ĐẶT MUA

-Chọn và đặt hàng



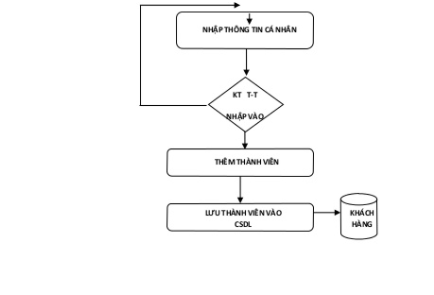
*Hinh14*

-Thanh toán



*Hinh15*

Đăng ký thành viên



Hinh16

### **2)Công cụ thực hiện**

-Visual Studio 2017

-My SQL 2016

## **PHẦN IV:TÀI LIỆU THAM KHẢO**

+Jose M. Garrido, “ASP.NET MVC Tutorial”, 6:20PM,

<https://dotnettutorials.net/course/asp-dot-net-mvc-tutorials/>

<https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx>

+ Entity Framework**:**

# <https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx>

+Tài liệu thầy Nguyễn Mai Huy, 2021,"bài giảng Lập Trình Web", khoa CNTT NTTU

.[truy cập 4/2021]

**KẾT LUẬN- HƯỚNG PHÁT TRIỂN**

**Ưu điểm**

+ Cơ bản nắm được các bước xây dựng một website bán hàng trực tuyến, thực hiện đúng quy trình

+ Xây dựng được bố cục trang web hợp lí, bước đầu thực hiện được nghiệp vụ của hệ thống

**Hạn chế**

+ website nhỏ, mang tính chất mô phỏng

+ Cơ sở dữ liệu nhỏ, chức năng phân quyền chưa tối ưu

+ Chưa kết hợp xử lí chức năng thanh toán hóa đơn

Hướng phát triển

+ Xây dựng website sử dụng được đáp ứng đầy đủ yêu cầu nghiệp vụ của hệ thống

+ Phân quyền và quản trị tối ưu

+ Cơ sở dữ liệu phong phú

+ Kết hợp chức năng thanh toán trực tiếp hay qua thẻ ATM(liên kết với hệ thống ngân hàng)