TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

Báo cáo Project II

**Đề tài: Xây dựng website quản lý kho**

**Sử dụng ASP.NET MVC**

Sinh viên thực hiện: **Nguyễn Ngọc Thành**

Lớp CNTT 1-2 K60

Giáo viên hướng dẫn: ThS. **Nguyễn Đức Tiến**

HÀ NỘI 05 -2018

# LỜI CẢM ƠN

Để có thể hoàn thành tốt môn học đồ án này, em xin gửi lời cảm ơn tới thầy giáo, ThS. Nguyễn Đức Tiến - Giảng viên bộ môn Kỹ thuật máy tính, viện Công nghệ thông tin và truyền thông, trường Đại học Bách Khoa Hà Nội đã tận tình hướng dẫn em trong quá trình làm đồ án.

Tuy nhiên, do thời gian và kiến thức có hạn nên đồ án này chắc chắn không tránh khỏi những thiếu sót, em rất mong được sự đóng góp ý kiến của thầy. Em xin chân thành cảm ơn.

Hà Nội ngày 23 tháng 5 năm 2018

Sinh viên: **Nguyễn Ngọc Thành**

# TÓM TẮT NỘI DUNG

Mục tiêu chính của đồ án là xây dựng một website quản lý kho có thể sử dụng trên mọi trình duyệt cũng như thiết bị có kết nối mạng. Ứng dụng này, giúp cho người dùng, chỉ với kết nối internet trên tay là có thể quản lý được kho hàng của mình.

Nội dung chính của đồ án gồm các phần:

**Chương I: Nền tảng và công nghệ phát triển website**

* Chương này tập trung đề cập đến nền tảng và công nghệ phát triển cho website, bao gồm những phần sau:
  + Tìm hiểu về mô hình MVC
  + Tìm hiểu về nền tảng ứng dụng web ASP.Net
  + Tìm hiểu về một số thư viện như Bootstrap, Bootbox, Datatables, …
  + Tìm hiểu về công nghệ web API, công nghệ bảo mật CSRF (Cross-Site Request Forgery), …
  + Tìm hiểu về một số ngôn ngữ lập trình Font – end như Html/Css/JavaScript,...
  + Tìm hiểu về SQL Sever và sự bá đạo của LinQ trong C#
  + Tìm hiểu về Entity Framewrok(EF)

**Chương II: Phân tích và thiết kế hệ thống**

* Vì thời gian có hạn nên website chỉ đầu tư về mặt Back - end nên chương này tập trung vào việc khảo sát, phân tích và thiết kế DataBase cho dự án.
  + Phần cuối là thiết kế hệ thống : Cấu trúc chương trình.

**Chương III: Tổng kết và đánh giá**

* Chương này tập trung vào việc đánh giá kết quả đạt được, đưa ra phương hướng phát triển cho hệ thống sau khi kết thúc đồ án và phần kết luận.

**LỜI NÓI ĐẦU**

Ngày nay, công nghệ thông tin được áp dụng rộng rãi trên mọi lĩnh vực. Máy tính điện tử không còn là phương tiện quý hiếm mà đã trở nên gần gũi với con người. Và việc sử dụng internet được sử dụng khắp nơi trong nước và quốc tế. Điều đó đã giúp mọi người trao đổi thông tin, trò chuyện, kết nối với nhau. Trước sự bùng nổ thông tin, các tổ chức, doanh nghiệp đều tìm mọi biện pháp để xây dựng hoàn thiện hệ thống thông tin cho mình nhằm tin học hóa các hoạt động tác vụ của mình. Việc nay, với việc áp dụng CNTT trong mô hình quản lý trực tuyến đã mang lại nhiều thuận lợi và hiệu quả: Tiết kiệm thời gian tra cứu kho hàng, thời gian quản lý kho, … Với lợi ích đó, chắc chắn nó là xu hướng phát triển mạnh mẽ trong tương lai. Trong xu thế đó, xây dựng một website quản lý kho là cần thiết. Chính vì vậy, e thực hiện đề tài:

**“Xây dựng website quản lý kho sử dụng ASP.NET MVC”**

.

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_Toc516177339)

[TÓM TẮT NỘI DUNG 3](#_Toc516177340)

[LỜI NÓI ĐẦU 4](#_Toc516177341)

[MỤC LỤC 5](#_Toc516177342)

[PHẦN I: NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN ỨNG DỤNG 7](#_Toc516177343)

[1.1. Tổng quan về ASP.NET 7](#_Toc516177344)

[1.2. Mô hình ASP.NET MVC 7](#_Toc516177345)

[1.3. Một số Framework Bootstrap, Bootbox.js, Datatables 8](#_Toc516177346)

[1.3.1. Bootstrap Framework 8](#_Toc516177347)

[1.3.2. Libraly Datatables 8](#_Toc516177348)

[1.3.3. Bootbox.js 8](#_Toc516177349)

[1.4. RESTful Service trong ASP.NET Web API 9](#_Toc516177350)

[1.5. CSRF – Cross-Site Request Forgery 10](#_Toc516177351)

[1.6. SQL Sever và LinQ 10](#_Toc516177352)

[1.7. Entity Framework(EF) 10](#_Toc516177353)

[PHẦN II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 12](#_Toc516177354)

[2.1. Phân tích đặc tả - Cơ sở dữ liệu 12](#_Toc516177355)

[2.2. Phân tích và thiết kế 14](#_Toc516177356)

[2.2.1. Biểu đồ Usecase 14](#_Toc516177357)

[2.2.2. Biểu đồ chức năng 15](#_Toc516177358)

[PHẦN III: CÀI ĐẶT VÀ TRIỂN KHAI 16](#_Toc516177359)

[PHẦN IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 19](#_Toc516177360)

C:\dir

# PHẦN I: NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN ỨNG DỤNG

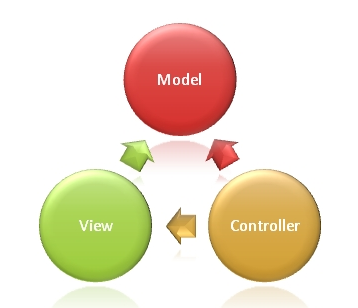
Trong chương này sẽ trình bày về nền tảng và các công nghệ trong việc xây dựng và phát triển website. Các vấn đề sẽ được trình bày gồm: Công nghệ ASP.NET, mô hình MVC, …

## Tổng quan về ASP.NET

* ASP.NET là một nền tảng ứng dụng web được phát triển và cung cấp bởi Microsoft, cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web. ASP.NET sử dụng .NET Framework, .NET Framework là sự tổng hợp tất các các kỷ thuật cần thiết cho việc xây dựng một ứng dụng nền desktop, ứng dụng web, web services...

## Mô hình ASP.NET MVC

* Mẫu kiến trúc Model – View – Controller được sử dụng nhằm chia ứng dụng thành ba thành phần chính: model, view và controller. Nền tảng ASP.NET MVC giúp cho chúng ta có thể tạo được các ứng dụng web áp dụng mô hình MVC thay vì tạo ứng dụng theo mẫu ASP.NET Web Forsm. Nền tảng ASP.NET MVC có đặc điểm nổi bật là nhẹ (lighweigt), dễ kiểm thử phần giao diện (so với ứng dụng Web Forms), tích hợp các tính năng có sẵn của ASP.NET. Nền tảng ASP.NET MVC được định nghĩa trong namespace System.Web.Mvc và là một phần của name space System.Web.
* Nền tảng MVC bao gồm những thành phần dưới đây:



Hình 1: Mẫu Model – View – Controller

* *Models*: Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ như, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server.
* *Views*: Views là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Product.
* *Controllers*: Controller là các thành phần dùng để quản lý tương tác người dùng, làm việc với model và chọn view để hiển thị giao diện người dùng. Trong một ứng dụng MVC, view chỉ được dùng để hiển thị thông tin, controller chịu trách nhiệm quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng. Ví dụ, controller sẽ quản lý các dữ liệu người dùng gởi lên (query-string values) và gởi các giá trị đó đến model, model sẽ lấy dữ liệu từ CSDL nhờ vào các giá trị này.

## Một số Framework Bootstrap, Bootbox.js, Datatables

### Bootstrap Framework

* Bootstrap là một framework HTML, CSS, và JavaScript cho phép thiết kế phát triển responsive web mobile một cách nhanh, gọn và dễ dàng hơn. Bootstrap bao gồm các HTML templates, CSS templates và Javascript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó.

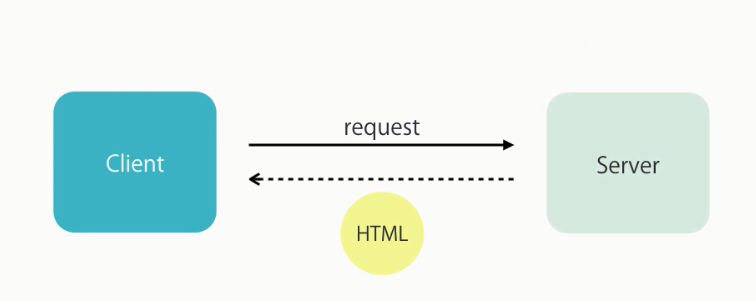
### Libraly Datatables

* DataTables là một plug-in cho thư viện jQuery Javascript. Nó là một công cụ tuyệt vời để thực hiện các công việc liên quan tới các bảng biểu được hiển thị trên các trang HTML. Với DataTables, việc Pagination, Seaching, Softing, .. sẽ vô cùng dễ dàng kể cả với những data lớn, đa dạng, hay các Array (List) lớn.

### Bootbox.js

* Bootbox.js là một thư viện nhỏ của javascript cho phép tạo các hộp thoại xuất hiện trên trình duyệt web khi người dùng sử dụng.

## RESTful Service trong ASP.NET Web API



Hình 2: Kiến trúc ASP.NET MVC

* Khi một client gửi một request tới Server, trong MVC Framework sẽ đến một Action trong Controller. Action này sẽ trả về một View, sau đó phân tích cú pháp, cuối cùng đánh dấu HTML (HTML Markup) và trả lại cho client. Vì vậy, trong cách tiếp cận này, HTTP Markup được tạo trên server và sau đó return cho client.
* Một cách khác để tạo HTML Markup. Vì không thể làm mọi thứ trên client, nên thay vì tới Action, hãy trả về Markup và Row Data (Dữ liệu thô). Cách này sẽ tiết kiệm được rất nhiều tài nguyên server; Row Data mở yêu cầu ít tài nguyên hơn. Vì vậy, dữ liệu có thể đến nhanh hơn cho client (Cải thiện hiệu suất hoạt động của ứng dụng). Và lợi ích cuối cùng là chúng ta có thể hỗ trợ cho một loạt các client, giống như mobile hay tablet app. Các ứng dụng này chỉ đơn giản là nhận dữ liệu và tạo chế độ xem cục bộ.
* Để gửi dữ liệu thô đi, chúng ta có hai cách là sử dụng Data Services và Web APIs (Application Programing Interface). Hiện tại, các web api không chỉ dừng lại ở việc xây dựng các ứng dụng dành cho thiết bị di động và máy tính bảng, mà các trang web lớn đều sử dụng web api để xây dựng chức năng mới mà không tốn nhiều tài nguyên lại có thể hiển thị dữ liệu công khai trong ứng dụng web như Facebook hay Twitter. Điều này giúp ta có thể hợp nhất dữ liệu của họ và dữ liệu của chúng ta và cung cấp trải nghiệm mới cho người dùng web.
* Trong bài tập lớn này, Framework em sử dụng để xây dựng RESTful – service là ASP.NET Web API và được xây dựng trên các nguyên tắc, cấu trúc tương tự ASP.NET MVC. Vì vậy, nó cũng có Routing, Controller, Action, ActionResult, … Với Web API, thay vì tạo HTML Markup cho hàng loạt client, ta có thể tạo một API Web để return tất các các client đó. Và sử dụng Jquery Plug – in: Datatables, để xây dựng bảng bên phía client có đầy đủ chức năng paginnation, seaching, sorting mà không cần viết thêm bất cứ một dòng code nào cho các chức năng đó.

## CSRF – Cross-Site Request Forgery

* Trong bảo mật thông tin, CSRF này gọi là giả mạo request khi chúng giả mạo một request không phải từ chính website mà hacker sẽ giả lập request để gửi các thông tin lên server mà không qua hệ thống website. Cho phép kẻ tấn công giả mạo thông tin user để thực hiện các hành vi ngoài mong muốn như ví dụ sau:
  + 1. Người dùng đăng nhập vào tài khoản ngân hàng
    2. Ngân hàng cấp quyền và một session bảo mật được khởi tạo giữa user và hệ thống ngân hàng
    3. Hacker gửi một thông điệp với link nói rằng bạn có thể kiếm 1.000.000.000 ngay bây giờ.
    4. Người dùng click vào link nguy hiển trên, sitesex chuyển tiền từ account của bạn về tài khoản của hacker. Bở vì session đó được tạo các đoạn mã đọc sẽ được thực thi thành công.
* Thêm một ví dụ vô cùng đáng sợ đã được phát hiện tại website bán vé xem phim của cụm rạp Lotte Cinema [tại đây](https://www.youtube.com/watch?v=CtnfOZmKR3A).
* Trong ASP.NET, Microsoft đã biết điều này và để ngăn chặn điều đó, ta chỉ cần gọi AntiForgeryToke

## SQL Sever và LinQ

* **SQL server** là ngôn ngữ phi thủ tục, không yêu cầu cách thức truy cập cơ sở dữ liệu như thế nào. Tất cả các thông báo của SQL rất dễ sử dụng và ít mắc lỗi. SQL cung cấp các tập lệnh phong phú đòi hỏi đáp dữ liệu như chèn, xóa, cập nhật các hàng trong một quan hệ. Tạo thêm, xóa, sửa một đối tượng trong một cơ sở dữ liệu. Điều khiển truy cập tới cơ sử dữ liệu và các đối tượng của cơ sở dữ liệu để đảm bảo tính bảo mật , tính nhất quán và sự ràng buộc của cơ sở dữ liệu.
* **LinQ** là một thư viện tích hợp trong .NET Framework, giúp quản lý database theo mô hình MVC và kết nối tới database theo dạng disconected. LinQ có hai cách sử dụng là Query Syntax và method syntax. Thư viện này cung cấp khả năng truy vấn database ngay trên ngôn ngữ lập trình. Khi sử dụng LinQ sẽ mô phỏng lại và lưu trong chương trình, sau đó mỗi Entity sẽ xây dựng thành một class như mô hình MVC từ đó ta có thể truy xuất database thông qua những class đó.

## Entity Framework(EF)

* Là một framework ánh xạ quan hệ đối tượng dành cho ADO.NET, là một 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



Hình 3: Vị trí của EF trong mô hình ASP.NET MVC

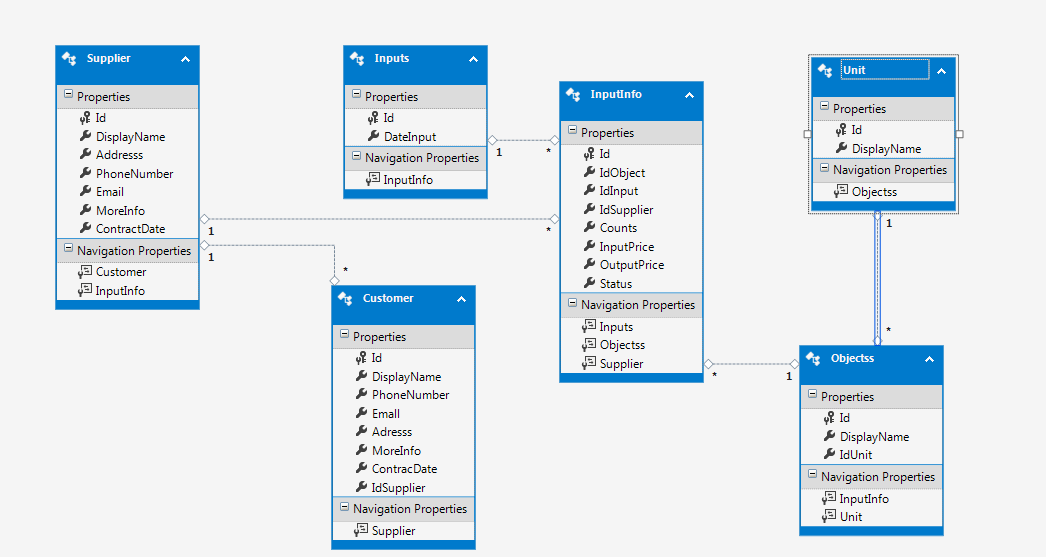
* EF là nơi trung gian để tương tác dữ liệu từ database tới các tầng cao hơn.
* Hệ thống web site trong dự án được xây dựng trên phương pháp code first mà Entity hỗ trợ. Tức là tạo đối tượng bảng cơ sở dữ liệu dựa trên model

# PHẦN II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

* **BKManager** là web site động , một hệ thống cung cấp khả năng quản lý kho hàng. Giúp cho Admin có thể dễ dàng quản lý, thêm xửa xóa sản phẩm, Suppler(khách hàng) cũng có thể theo dõi mặt hàng của mình.
* **Thành phần**: admin và khách hàng.
* **Giao diện**: - trang chủ, đăng ký, đăng nhập; trang quản lý cho admin; trang cho supplier.
* **Mô trường làm việc**: Visual Studio 2017
* **Công nghệ áp dụng**: ASP.NET MVC; Html/css/js…
* **Phân quyền**: Admin có quyền thêm sửa xóa sản phẩm, xem các order; khách hàng có quyền xem sản phẩm..
* **Phạm vi**: phục vụ cho cửa hàng, đại lý, nhà cung cấp.

## Phân tích đặc tả - Cơ sở dữ liệu

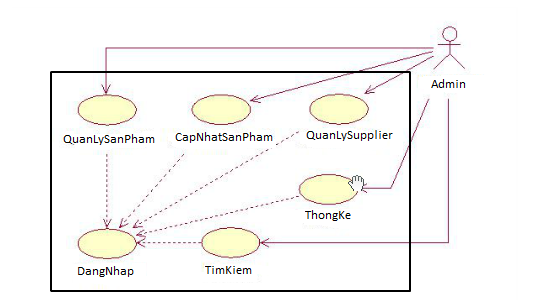
* Phần mềm quản lý kho, ta phải quản lý việc: Đọc, thêm, xóa, sửa (CURD) object. Object ở đây là các mặt hàng có trong kho mà ta cần quản lý. Những đối tượng cần quan tâm:
  + 1. Object (Vật phẩm – Mặt hàng có trong kho):
       - Id
       - DisplayName (Tên hiển thị): Để phân biệt với Name, PublicName, PrivateName, ortherName,…
       - UnitId: Khóa ngoại tham chiếu tới bảng Unit
       - InputId: Sản phẩm này được nhập vào từ phiếu nhập nào? Trong phiếu nhập đã có sẵn supplier cung cấp nên ta cũng không cần thêm khóa ngoại tới Supplier table ở đây.
    2. Unit ( = Stock – Đơn vị tính của đối tượng):
       - Id
       - DisplayName
    3. Supplier (Nhà cung cấp)
       - Id
       - DisplayName
       - Address
       - Phone
       - Email
       - ContractDate: Ngày hợp tác
       - MoreInfo
    4. Customer: Khách hàng của nhà cung cấp nào? Phục vụ việc support cho khách hàng sau này.
       - Id
       - DisplayName
       - Address
       - Phone
       - Email
       - MoreInfo
       - SupplierId: Khách hàng của nhà cung cấp nào?
    5. Input: Phiếu nhập (Cần phải biết nhập cái gì? Thời gian nhập? Sản phẩm đó nhập của ai? Số lượng nhập? Tình trạng nhập vào như thế nào? Giá nhập vào bao nhiêu? Từ Supplier, họ sẽ cung cấp hàng cho mình, khi đó mình sẽ nhập vào. Ta có thêm một vài đối tượng để tiện dụng cho việc cấp trên xem xét sau này, hoặc truy vấn danh sách nhập/xuất trong 1 khoảng thời gian bất kì nào đó hoặc tiến hành bảo hành sản phẩm cho khách hàng.)
       - Id
       - DisplayName: Tên phiếu nhập
    6. InputInfo (Thông tin phiếu nhập): Nếu chỉ có một bảng Input, ta chỉ có thể lưu thông tin nhập của 1 sản phẩm, trong 1 lúc trong 1 phiếu nên trong trường hợp một nhà cung cấp cung cấp nhiều sản phẩm, trong cùng 1 lúc, nếu chỉ có 1 bảng input sẽ không đáp ứng được. Ta cần thêm 1 bảng để lưu cho từng mặt hàng.
       - Id
       - InputId: Sản phẩm nằm trong phiếu nhập nào?
       - ObjectId: Nhập cái gì?
       - SupplierId: Nhập của ai?
       - Count: Nhập vào bao nhiêu?
       - InputPrice: Giá nhập vào như thế nào?
       - OutputPrice: Giá bán ra lúc đó là bao nhiêu?



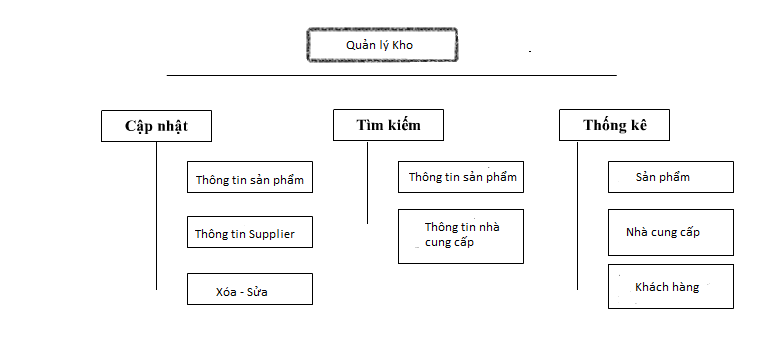
Hình 4: Table và lược đồ quan hệ RelationShips

## Phân tích và thiết kế

### Biểu đồ Usecase

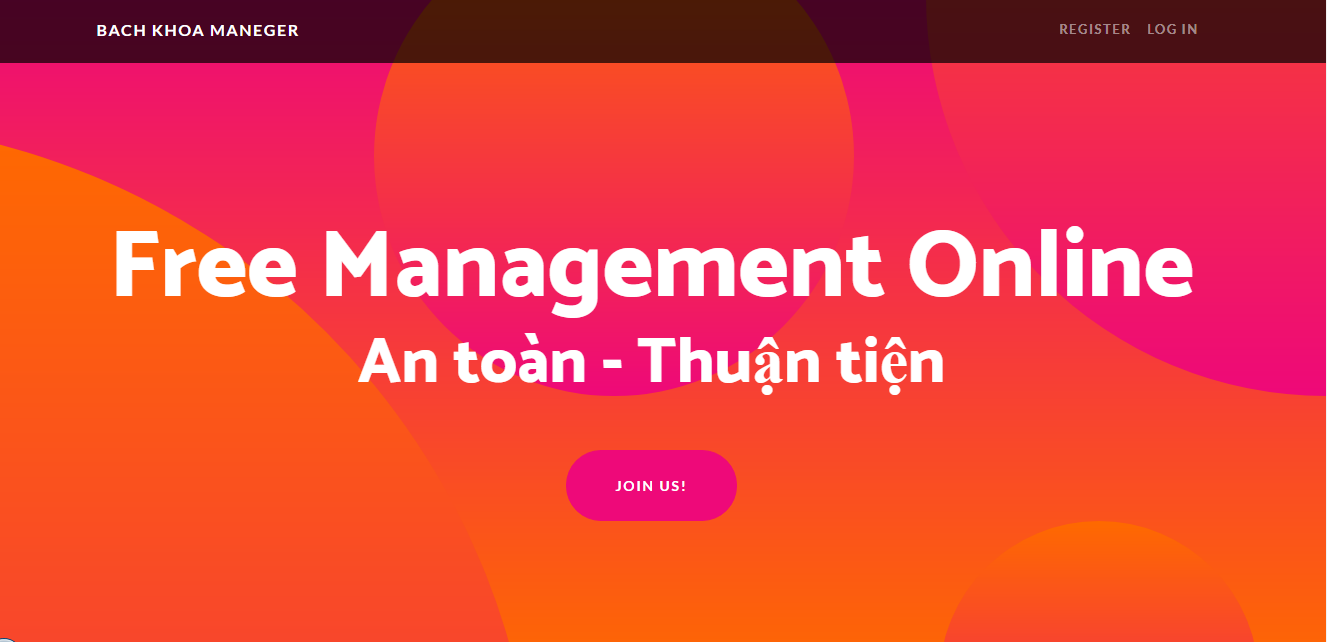


### Biểu đồ chức năng

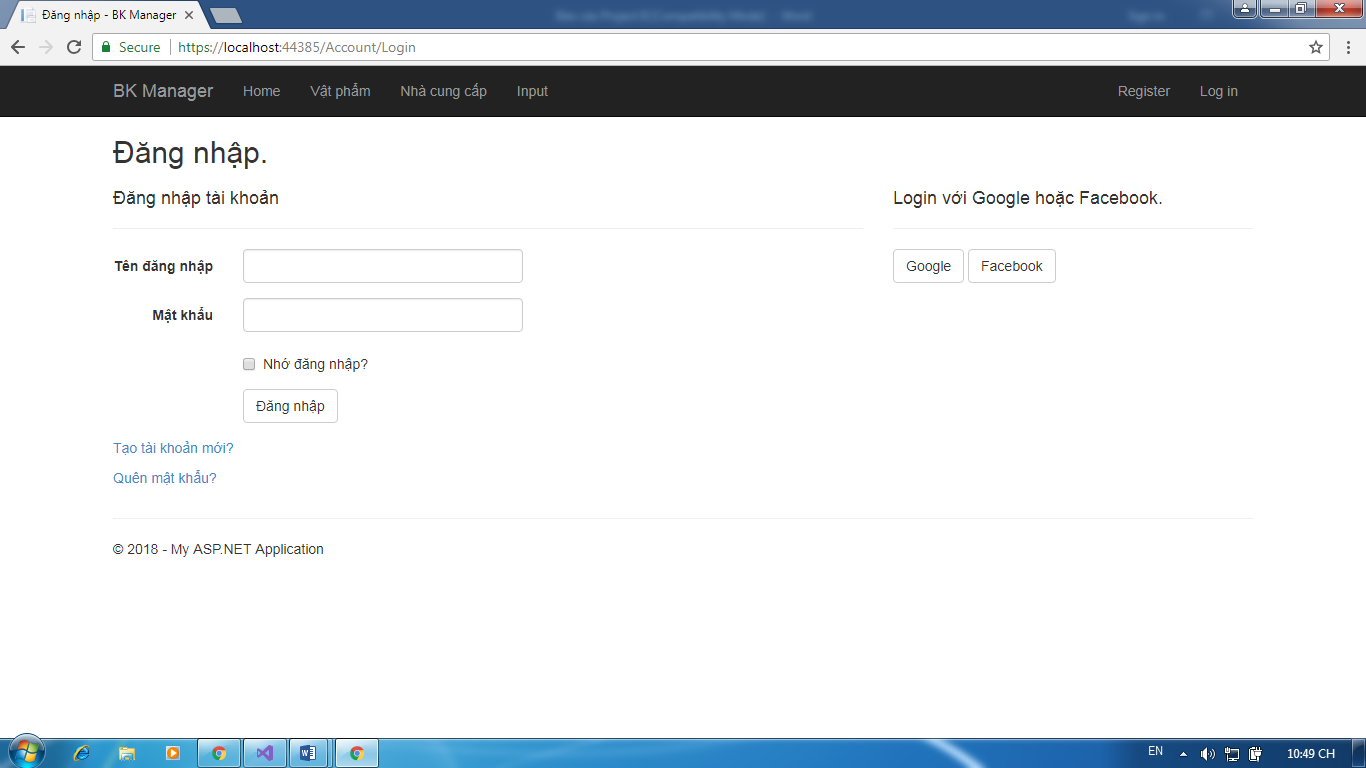


# PHẦN III: CÀI ĐẶT VÀ TRIỂN KHAI

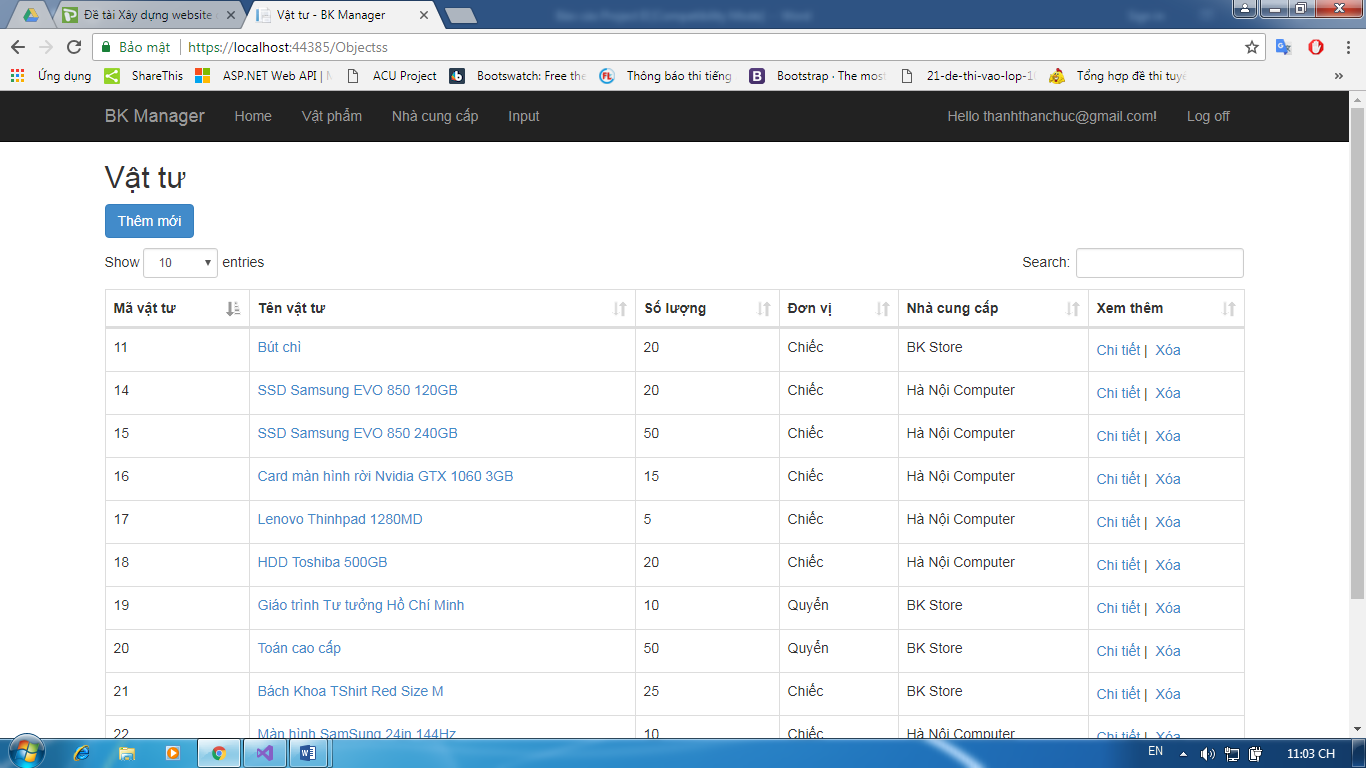
* Form ban đầu



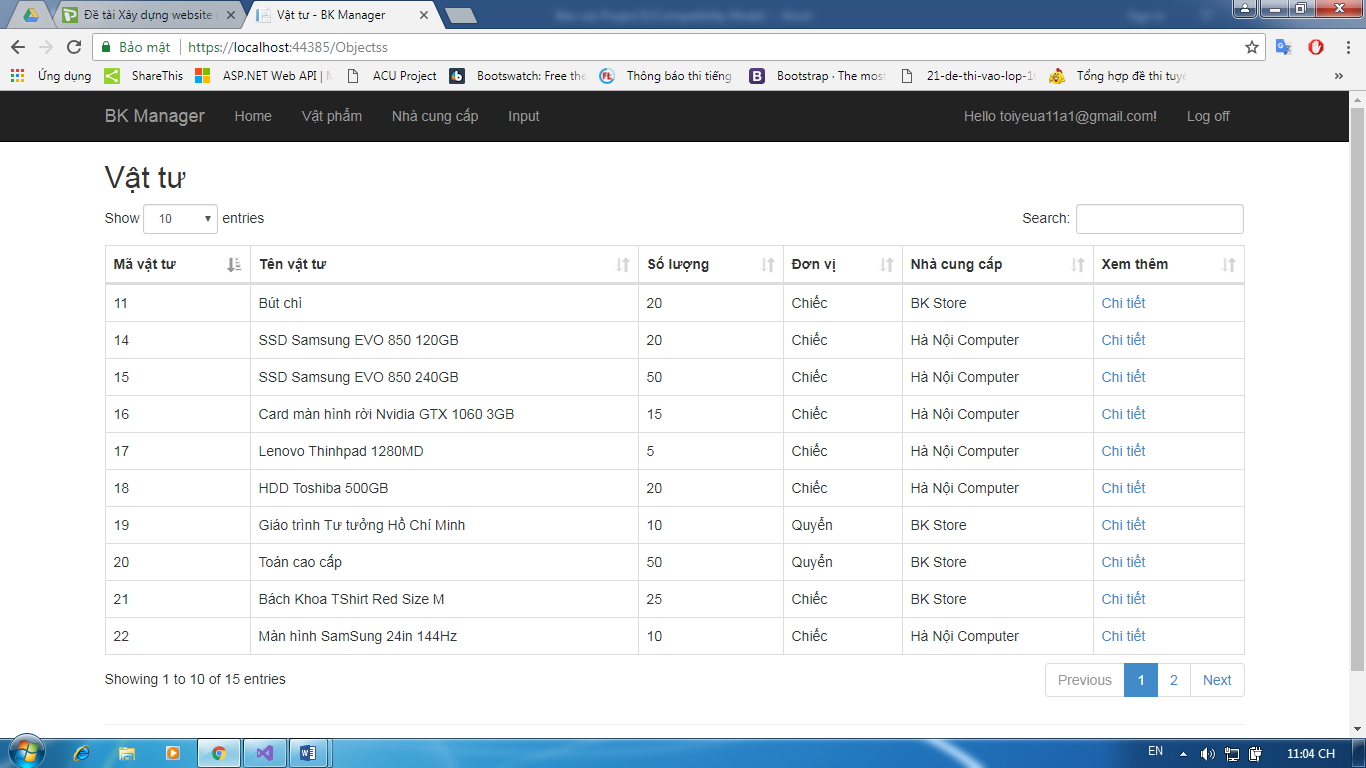
* Form Login



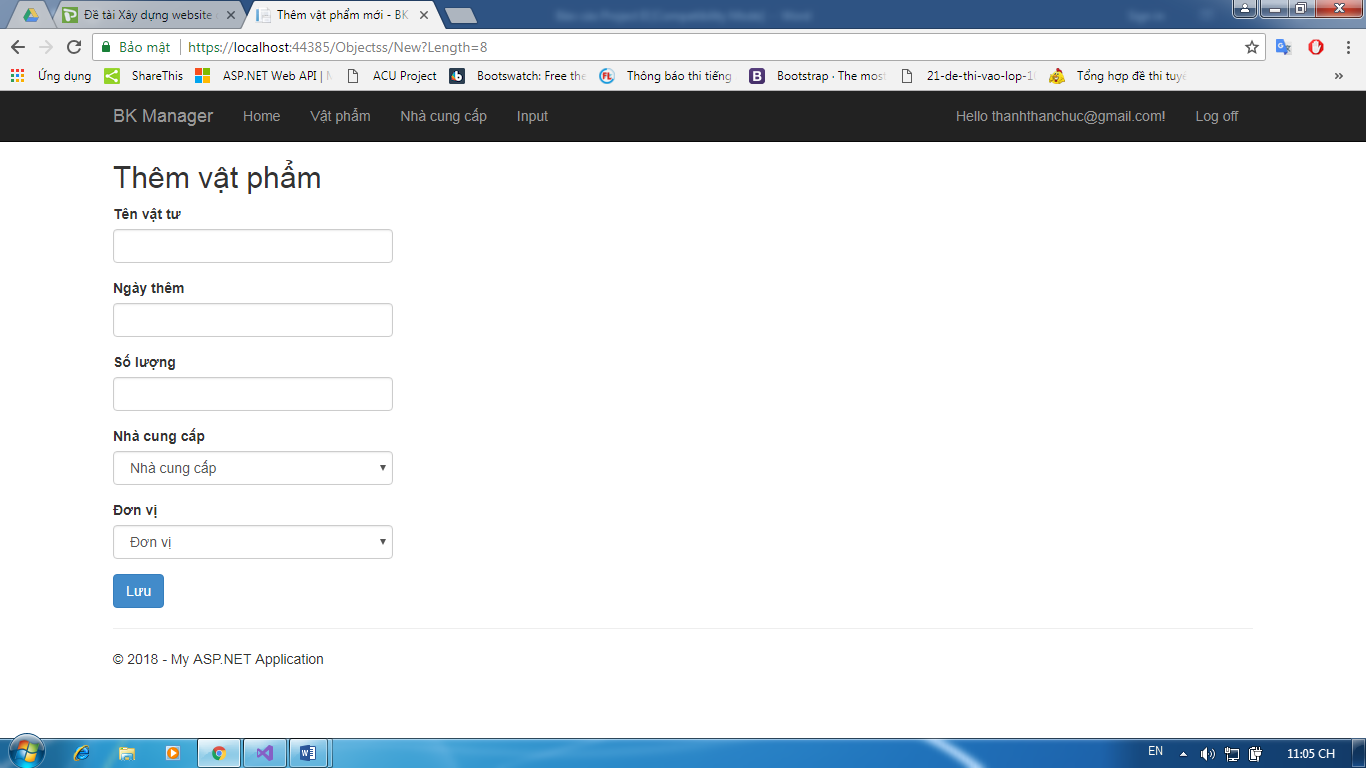
* Form Vật phẩm của Admin



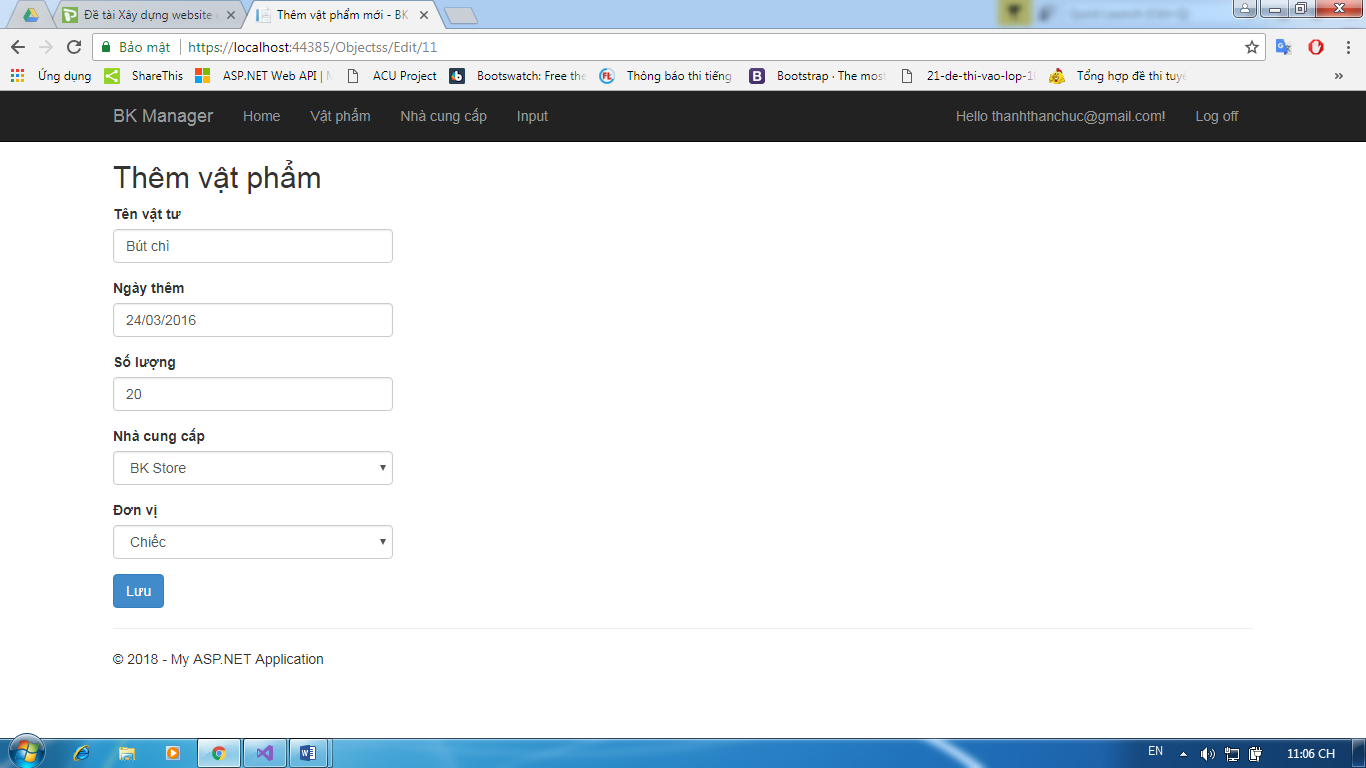
* Form vật phẩm của Supplier



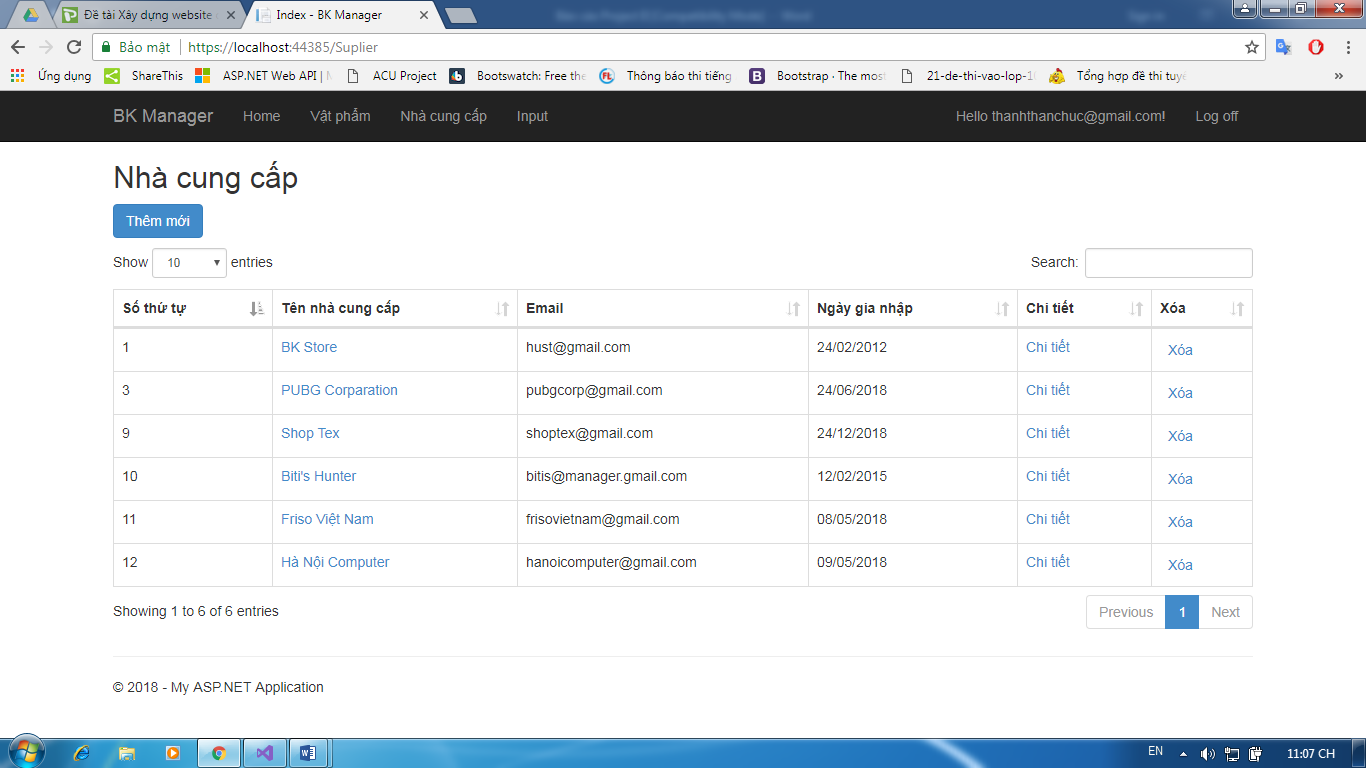
* Form tạo mới vật phẩm



* Form chỉnh sửa sản phẩm



* Form danh sách khách hàng(Supplier)



* Và nhiều form chức năng khác.

# PHẦN IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* 1. **Những kết quả đạt được**
* Đề tài: “Xây dựng website quản lý kho” bap gồm các phần: Phân tích thiết kế hệ thống, xây dựng cơ sở dữ liệu và lập trình xây dựng chương trình.
* Ưu điểm:
  + - * Về mặt lý thuyết: Đã nghiên cứu và tìm hiểu được quy trình phân tích, thiết kế hệ thống, tìm hiểu về nhiều công cụ, ngôn ngữ như C#, Asp.NET, MySQL, …, áp dụng thành thông vào 1 bài toán thực tế, xây dựng website quản lý kho hàng.
      * Về mặt thực hành: Đã xây dựng được một website quản lý hàng hóa thực hiện các chức năng: Thêm, xóa , sửa, xem sản phẩm, nhà cung cấp, thống kê, tìm kiếm, sắp xếp sản phẩm, …
* Nhược điểm:
  + - * Đề tài xây dựng chỉ là một phần nhỏ trong hệ thống tin học quản lý, chưa đáp ứng đầy đủ yêu cầu của một bài toán quản lý hoàn chỉnh.
      * Hệ thống xây dựng chỉ quan tâm đến mặt Back-End nên giao diện còn chưa được đẹp mắt.
  1. **Hướng phát triển đề tài**
* Phát triển hệ thống quản lý, cho phép mỗi supplier có thểm thêm, xóa, sửa chính sản phẩm của họ khi đăng nhập thành công vào website.
* Bổ xung và hoàn thiện thêm các chức năng quản lý được chặt chẽ hơn.

TÀI LIỆU THAM KHẢO

***\* Danh mục website:***

1. https://msdn.microsoft.com/en-us/library/aa286485.aspx.
2. https://bachkhoa-aptech.edu.vn/huong-dan-su-dung-entity-framework-code-first-trong-asp-net-mvc.htmliOS Technology Overview,
3. https://www.stdio.vn/articles/linq-gioi-thieu-va-cach-su-dung-95Stack Overflow, <http://stackoverflow.com/>.
4. Youtube.com
5. http://google.com