ĐỒ ÁN CUỐI KÌ MÔN PHÁT TRIỂN ỨNG DỤNG WEB

*Website bán hàng online*

# Thông tin nhóm thực hiện

## Thông tin cơ bản

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | MSSV | Họ tên | Email | Điện thoại | Điểm đề nghị |
| 1 | 1312032 | Nguyễn Hữu Bảo | [huubao2309@gmail.com](mailto:huubao2309@gmail.com) | 0979313803 |  |
| 2 | 1312120 | Hà Minh Đạt | [haminhdat01@gmail.com](mailto:haminhdat01@gmail.com) | 0983522369 |  |

## Phân chia công việc

### 1312032 – Nguyễn Hữu Bảo

* Phụ trách thực hiện các chức năng từ phía server side:
* Thiết kế, tạo CSDL và tối ưu CSDL
* Triển khai Repository
* Tạo tầng Service
* Viết các Unit Test cho tầng Service này (Reporsitory, Service, Web Api, vv…)
* Triển khai ASP.NET Identity [1]
* Phụ trách phân trang quản trị:
* Dựng Template và trang master
* Viết các chức năng đăng nhập, đăng xuất, vv…
* Quản lý thông tin User
* Quản lý danh mục sản phẩm (Thêm, xóa, sửa và đọc dữ liệu).
* Quản lý danh sách sản phẩm (thêm tin tức, quản lý ảnh, vv…)

### 1312120 – Hà Minh Đạt

* Phụ trách phần giao diện cho khách hàng đăng nhập:
* Hiển thị danh sách sản phẩm
* Hiển thị trang chủ
* Tạo danh sách sản phẩm
* Thực hiện chức năng tìm kiếm
* Đăng nhập và đăng xuất của khách hàng
* Đăng nhập với Google+ [2] và Facebook [3]
* Gửi mail cho khách hàng
* Thực hiện chức năng thống kê truy cập
* Chức năng giỏ hàng và thanh toán qua Ngân Lượng [4]

# Thông tin chung về đồ án

* Tạo một website bán hàng trực tuyến với các chức năng sau:
* Khách hàng xem, tìm kiếm và mua sản phẩm
* Khách hàng có thể đăng nhập và xem danh sách đơn hàng đã mua
* Khách hàng có thể gửi liên hệ và xem thông tin công ty
* Admin quản lý sản phẩm, đơn hàng, tin bài và nhận xét của người dùng.
* Khách hàng có thể đăng nhập hoặc không đăng nhập, việc đăng nhập sẽ giúp khách hàng quản lý giỏ hàng, thanh toán, vv…
* Admin thì phải bắt buộc phải đăng nhập.

# Thuyết minh nội dung

## Phát biểu vấn đề

* Trong thời đại mà thương mại điện tử ngày càng phát triển nhanh như hiện nay, thì việc nhiều người muốn có website để kinh doanh, buôn bán trong lĩnh vực này thì ngày cảng tăng cao. Chính vì lí do đó, nhóm quyết định thực hiện một dự án là website bán hàng online.

## Mục tiêu đồ án

* Ngoài việc thực hiện đồ án để nắm các kiến thức căn bản, đồng thời nâng cao, tìm hiểu các kiến thức mới hơn. Thì mục tiêu khác của nhóm là thử tạo một website bán hàng online hoàn chỉnh để dự phòng cho sau này có thể sử dụng.

## Ý nghĩa thực hiện

* Tạo một website bán hàng online giúp cho chủ website có thể quản lý được sản phẩm của website của mình, xem lượng người truy cập website thường xuyên để đưa ra hướng đi cho việc phát triển website, cũng như dịch vụ của mình.
* Giúp giảm thiểu tối đa thời gian cho khách hàng trong việc mua sắm sản phẩm, thanh toán, vv…

# Nội dung triển khai

## Kiến trúc đồ án

* Đi theo cấu trúc sau:
* Database => Repository => Service => Web Api, Controller, RESTFUL API => AngularJS, ASP.NET, MVC, Bootstrap, vv….

## Công nghệ sử dụng

* Database: SQL server 2012
* Truy xuất dữ liệu: Entity Framework
* Server: ASP.NET
* Front-end: AngularJS, HTML, CSS, JS, Bootstrap
* Chứng thực: ASP.NET Identity [1]
* Quản lý source code: Github
* Host sử dụng: AppHarbor [5]
* Cloud tải hình: Cloudinary [6]

# Dự kiến kết quả

* Tạo một website bán hàng giúp cho khách hàng có thể mua sắm, tìm kiếm, mua, đặt hàng trên website, vv…
* Website giúp cho admin có thể quản lý danh sách sản phẩm, thêm, xóa, sửa, vv..
* Bảo vệ thông tin khách hàng, cũng như của admin.
* Thanh toán thông qua website Ngân Lượng [4].
* Thống kê lượng người dùng trang web.

# Các khó khăn gặp phải và cách giải quyết

## Khó khăn

* Khó khăn trong việc tìm hiểu các framework như AngularJS, JQuery, Ajax, Bootstrap, vv…
* Khó khăn trong việc tìm hiểu về cách làm việc trong Entity Framework.
* Gặp phải các lỗi liên quan đến Logic khi xử lý các chức năng phân quyền, bảo mật thông tin User, giỏ hàng, vv…
* Khó khăn trong việc thiết kế website cho trang admin và client.
* Gặp phải vấn đề trong việc upload source lên host AppHarbor [5].
* Lỗi trong quá trình upload ảnh giữa server và client.
* Gặp phải lỗi trong việc gắn chức năng thanh toán qua Ngân Lượng [4] trong website.
* Khó khăn trong việc làm việc nhóm, chia sẻ source code qua Github nhóm, nhiều lúc merger, push, pull, commit bị lỗi.

## Cách giải quyết

* Code đi code lại các bài tập về các module của framework, các bài tập đơn giản như web tính toán đơn giản, vv… để làm quen với cách code và làm việc với các framework như AngularJS, JQuery, Ajax, Bootstrap, vv…
* Tham khảo các tài liệu, video chỉ dẫn về cách làm việc với Entity Framework.
* Tham khảo các code mẫu về các thao tác làm việc về cơ chế bảo mật, phân quyền, giỏ hàng trong các website.
* Để thiết kê website, nhóm quyết định sử dụng các template giao diện có sẵn trên mạng, vừa đẹp, vừa đã xử lý sẵn bootstrap và responsive. Như trang admin thì có website [7] và trang client thì có website [8].
* Gặp phải vấn đề trong việc upload host, do AppHarbor có chơ chế ghi log lỗi khá tốt nên nhóm sửa dần dần cũng được.
* Vấn đề không hiển thị ảnh khi chạy web do lỗi úp ảnh thì nhóm dùng cloud chứa ảnh là cloudianry [6] để chứa ảnh giữa client và server.
* Xem video, hướng dẫn về tích hợp thanh toán qua Ngân lượng trong ASP.NET.

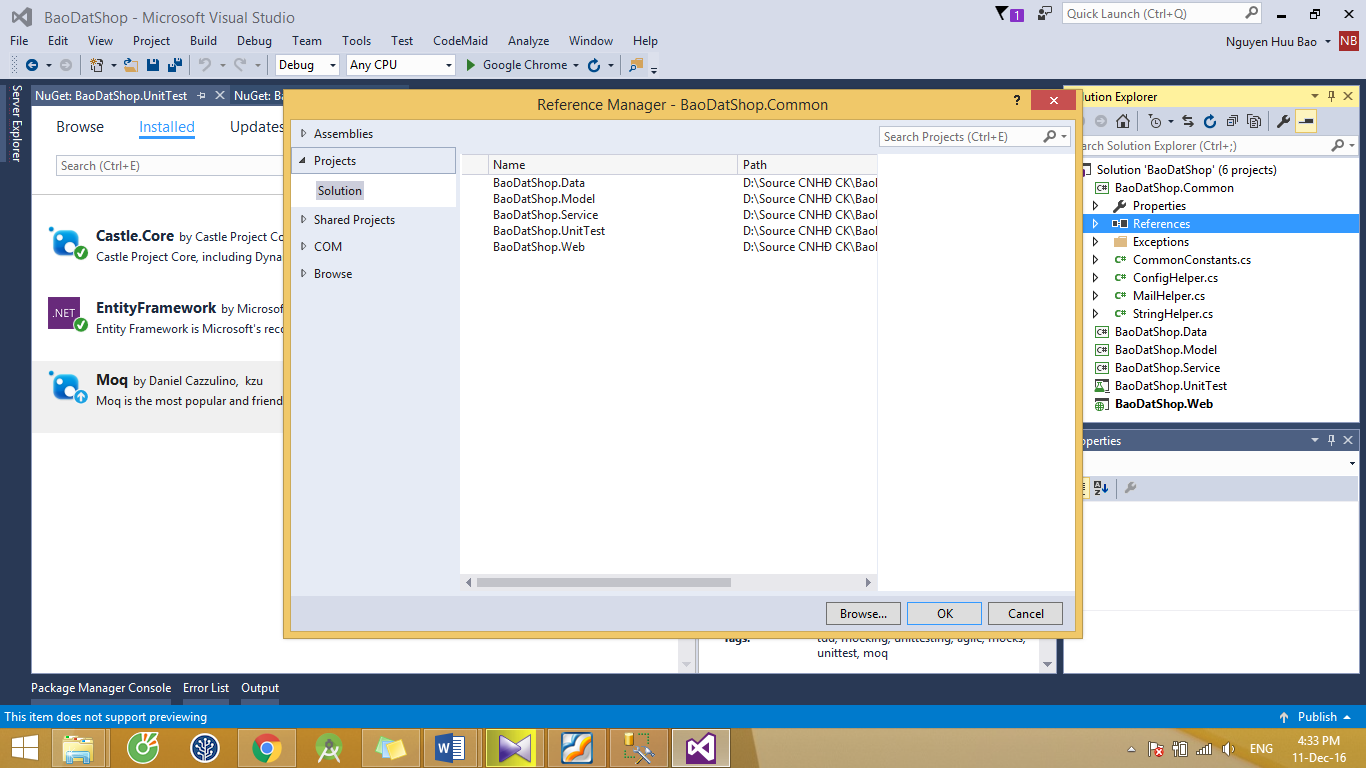
# Kiến trúc hệ thống

## Cấu trúc thư mục:

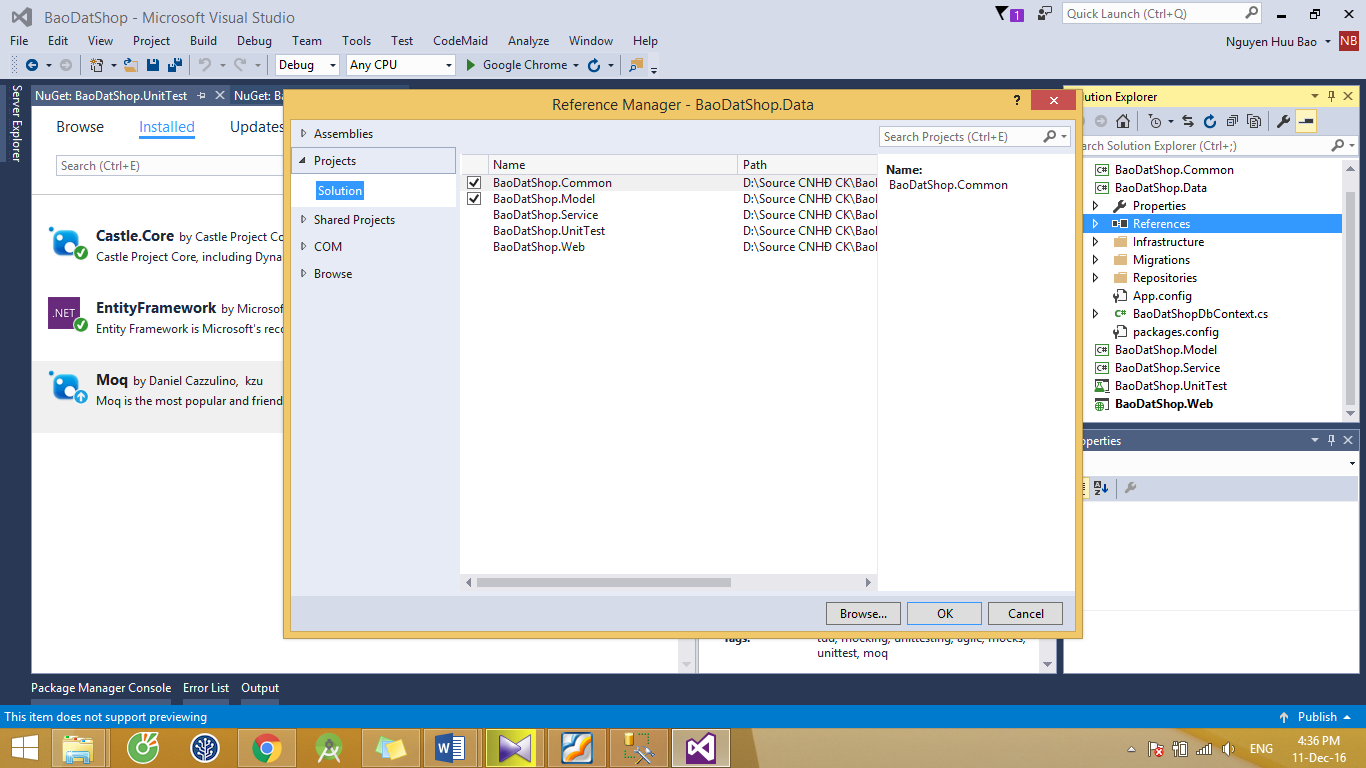
* **BaoDatShop.Common**: Chứa các lớp tiện ích dùng chung cho dự án
* **BaoDatShop.Model:** Chứa các lớp Domain Entities của dự án
* **BaoDatShop.Data**: Chứa tầng truy cập dữ liệu sủ dụng Entity Framework Codefirst
* **BaoDatShop.Service**: Chứa các service xử lý Business logic
* **BaoDatShop.Web**: Project chính dùng để hiển thị giao diện và tương tác người dùng.
* **BaoDatShop.UnitTest**: Chứa các class Text sử dụng cho việc Unit Test.

## Liên kết giữa các thư mục

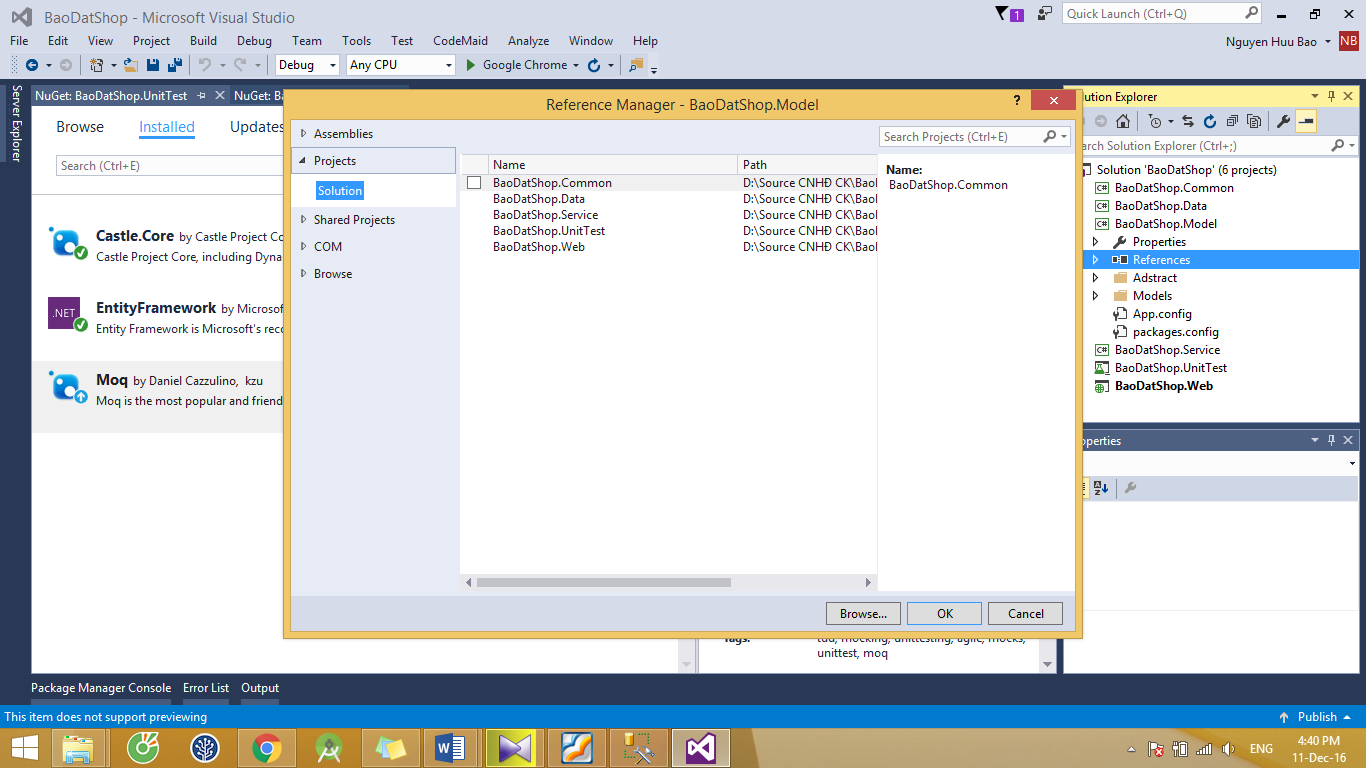
* **BaoDatShop.Common** không liên kết với bất kì thư mục nào



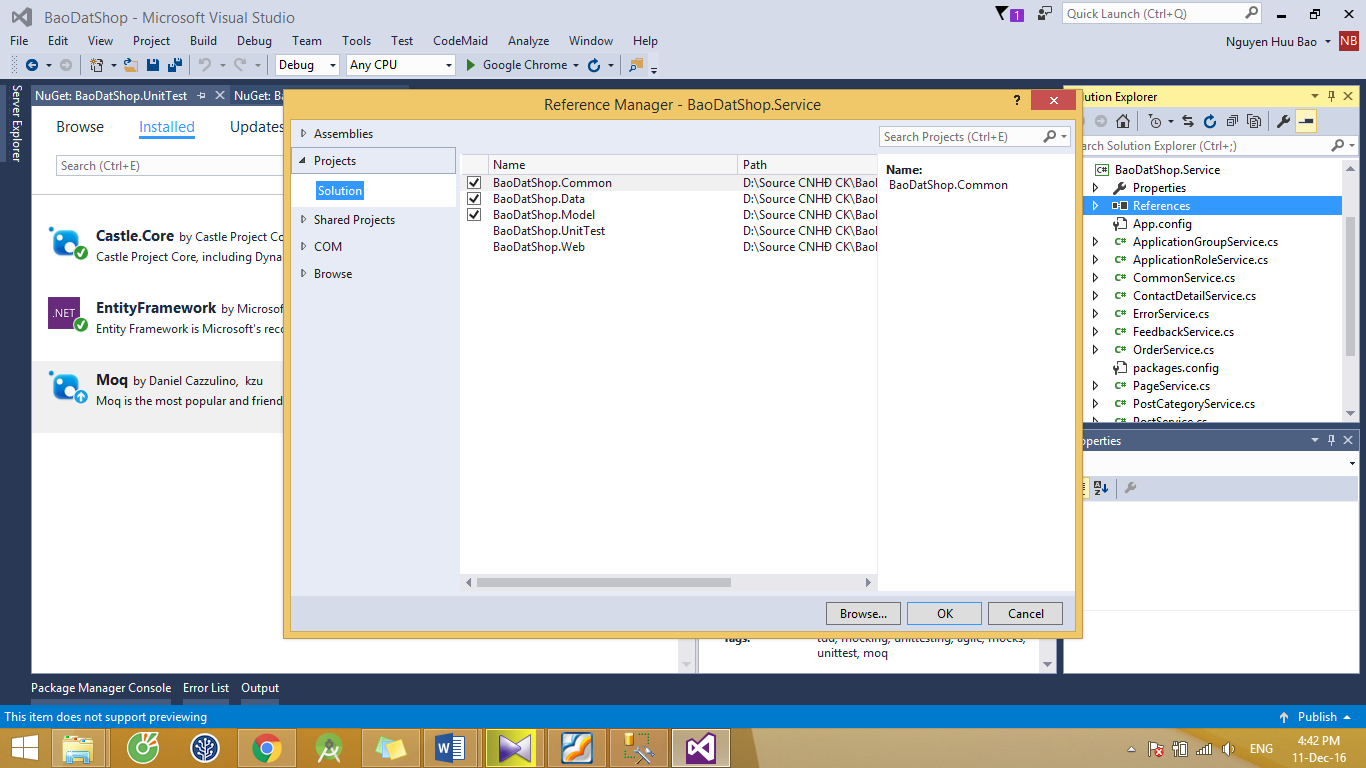
* **BaoDatShop.Data:** liên kết với các thư mục



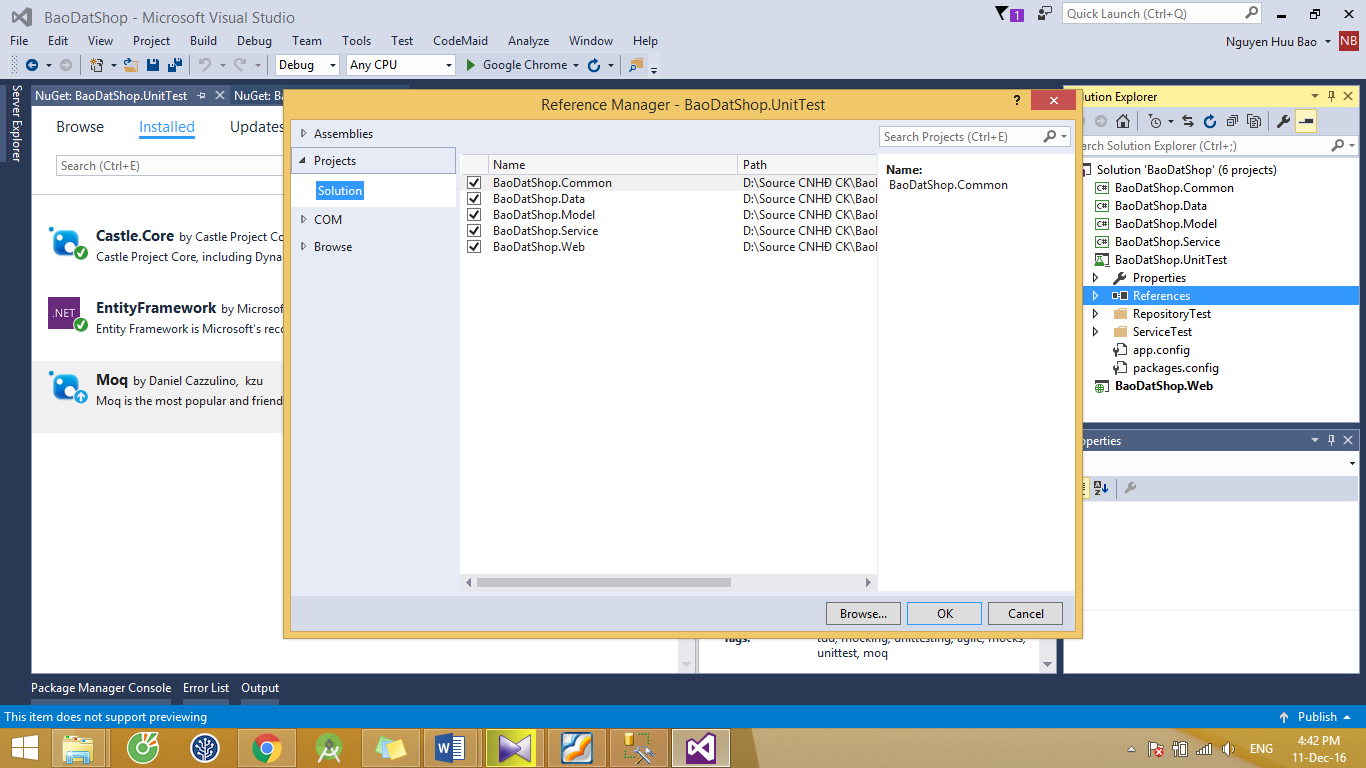
* **BaoDatShop.Model**: liên kết với các thư mục



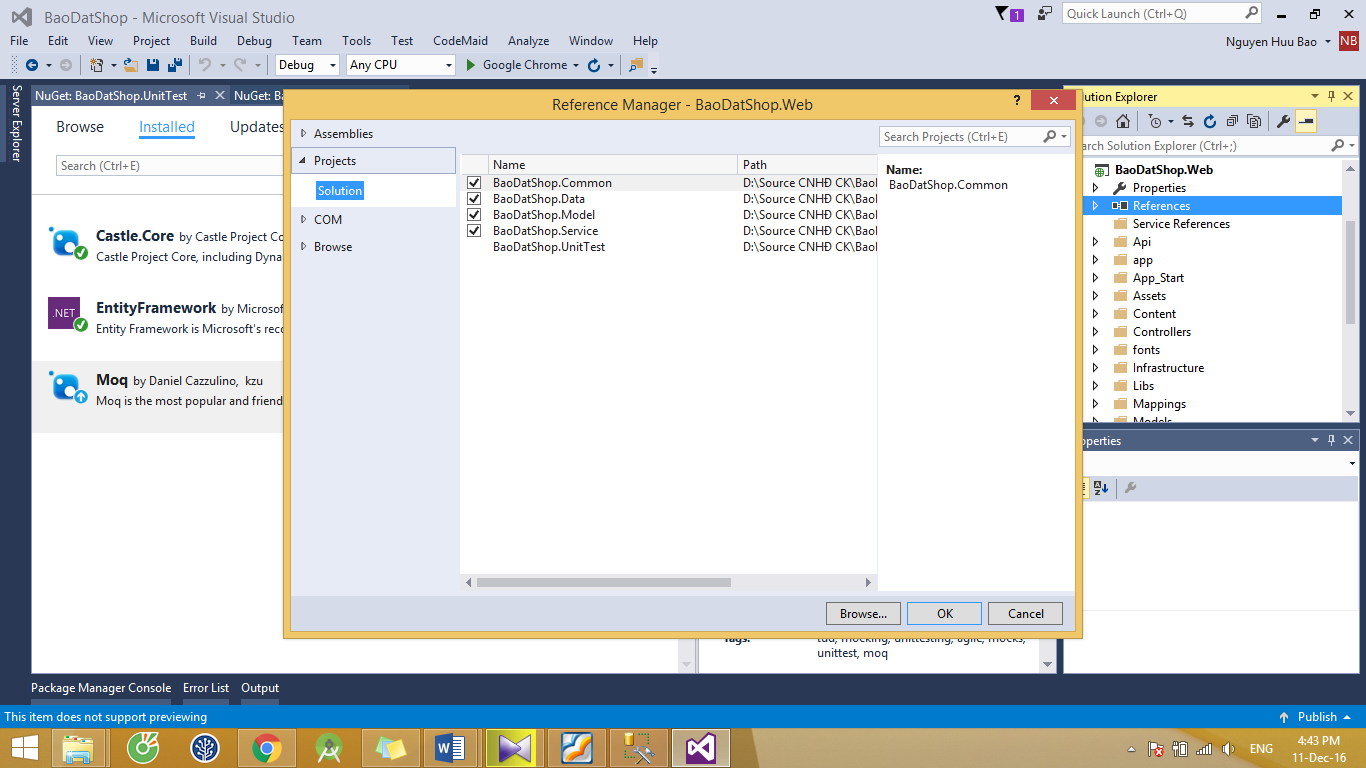
* **BaoDatShop.Service:** liên kết với các thư mục



* **BaoDatShop.UnitTest:** liên kết với các thư mục



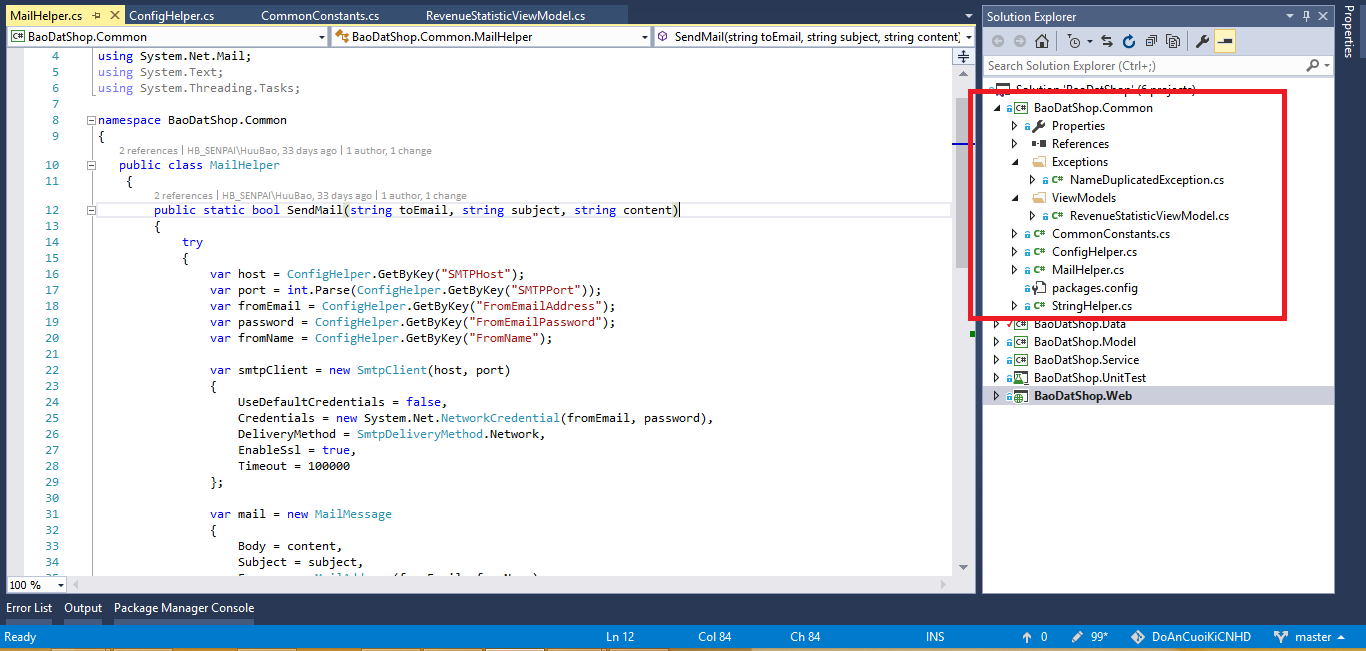
* **BaoDatShop.Web:** liên kết với các thư mục



## Chi tiết từng thư mục trong dự án

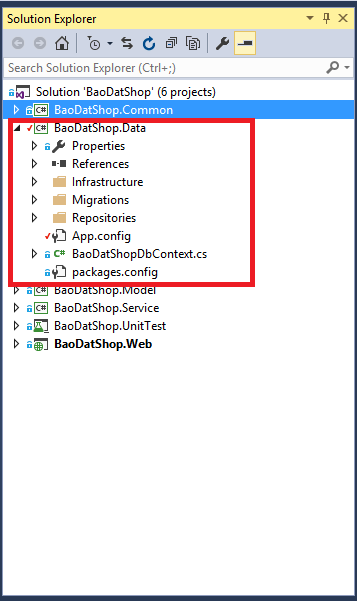
### BaoDatShop.Common

* Thư mục này chứa các lớp tiện ích dùng chung, hỗ trợ cho xử lý:

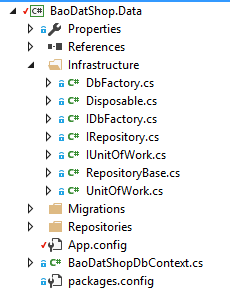


* Như các lớp **StringHelper.cs** để hỗ trợ xử lý, chuyển hóa kí tự.
* Lớp **ConfigHelper.cs** để hỗ trợ lấy các key trong file **AppSettings.config** trong thư mục **BaoDatShop.Web**.
* Lớp **MailHelper.cs** để hỗ trợ gửi mail chứng nhận đăng kí, chứng nhận thanh toán.
* Lớp **CommonConstants.cs** để hỗ trợ lấy định danh các tittle để làm SEO.
* Lớp **RevenueStatisticViewModel.cs** để tạo các đối tượng để thống kê doanh thu.

### BaoDatShop.Data



#### Thư mục Infrastructure tức là chứa hạ tầng (chuyên để chứa các Interface).



* 3 Interface: **IDbFactory (**Để giao tiếp khởi tạo các đối tượng**), IRepository, IUnitOfWork:** giao tiếp với database.
* Interface **Disposable** để hủy đối tượng và thu hồi vùng nhớ khi chúng ta cập nhật bảng dữ liệu.
* Interface IRepository để định nghĩa các phương thức khi chúng ta migration database
* Class DbFactory để kế thừa 2 class Disposable, IDbFactory.
* Class **UnitOfWork** để kế thừa IUnitOfWork để đẩy dữ liệu vào database
* Class RepositoryBase để kế thừa IRepository generate các đối tượng.

#### Thư mục Repositories để chứa các Repository:

* Các class lớp này kế thừa từ class IRepository và RepositoryBase.
* Trái tim của Entity Framework là class **BaoDatShopDbContex**t**.cs**. Sẽ khởi chạy khi khởi chạy Entity Framework. Có các đối tượng giống với trong CSDL.

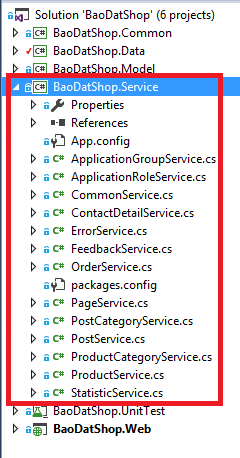
#### Thư mục Migrations:

* Thư mục này sẽ hiện file tóm tắt các lần add migration, khi ta chạy câu lệnh “***add-migrations Tên lần add***”.
* Các file này để check các lần add migartions.
* File **Configuration.cs** sẽ khởi chạy khi chúng ta migration (hàm Seed() để chạy), chúng ta có thể khởi tạo dữ liệu mẫu trong file này.

### BaoDatShop.Model

* Lớp Model này để tạo ra các model tương ứng với các bảng trong CSDL.
* Tách riêng lớp Model này riêng với lớp Model trong thư mục **BaoDatShop.Web.** Vì thư mục Model trong **BaoDatShop.Web** chỉ để thao tác với giao diện (View) thông qua Controller theo mô hình MVC. Còn lớp Model trong thư mục BaoDatShop.Model này chỉ để thao tác với Database.

### BaoDatShop.Service



* Vì tầng repository chỉ xử lí các câu lệnh common (chung) của 2 bảng chứ nếu thao tác trên 3 hay 4 bảng liên hệ với nhau. Nên ta cần phải tạo tầng service.
* Ngoài ra còn để thuận tiện cho việc các API và controller gọi lên, nên ta tách ra tầng service cho nó tiện.
* Môt class sẽ thao tác trên nhiều 2 hay nhiều bảng. **VD**: bảng Post và PostTag chúng ta sẽ tạo chung 1 service là PostService để thao tác những gì liên quan tới 2 bảng này. Trong class tạo interface IPostService để có các method làm việc giữa 2 bảng. Ví dụ hàm này:

|  |
| --- |
| public IEnumerable<Post> GetAll()  {  return \_postRepository.GetAll(new string[] { "PostCategory" });  } |

* Khi chúng ta thực thi hàm này thì nó sẽ get ra cả post và postcatelogy tương ứng.

### BaoDatShop.UnitTest

* Khi tạo unit test trong visual sẽ tạo sẵn MS Unit Test chúng ta chỉ việc thêm thư viện Moq, Chúng ta test cho 2 tầng: Repository và Service (2 thư mục RepositoryTest và ServiceTest).
* Để chạy các test unit thì chúng ta chọn trên thanh menu lần lượt: Test->Windows->Test Explorer. Sẽ xuất hiện danh sách các test case và chúng ta nhấn Run all 1 phát nó sẽ chạy tất cả các Test case.

### BaoDatShop.Web

* Thư mục **Asset** để chứa các tài nguyên của adim riêng và client riêng.
* Thư mục **Infrastructure** để chứa các class dùng chung cho website. Mục này chứa 2 thư mục là **Core** và **Extension**. Thư mục Core thì chứa class ApiControllerBase, sau này lớp nào muốn sử dụng API đều phải khai báo nó. Lớp này để thông báo lỗi và các reponse trả về.
* Thư mục **Controllers** để chứa các controllers để thao tác với các API (đường dẫn thư mục) của trang admin.
* Thư mục **API** thì để chưa các controllers để thao tác với các API (đường dẫn thư mục) của trang user.
* Thư mục **app/components** để chứa các class controller và view để làm việc với trang user.
* Thư mục **app/shared** để chứa các lớp hỗ trợ cho các thao tác làm việc với client (bao gồm trang user và admin)
* Thư mục **App\_Start** để chứa các file hỗ trợ việc cấu hình đường dẫn, token, kết nối facebook, vv…
* Thư mục **Mappings** chưa class **AutoMapperConfiguration.cs** để thêm vào phần Mapping giữa 2 bảng với nhau, hoặc mapping model bên ngoài và model bên trong thư mục Web.
* Thư mục **Views** để tạo ra các View cho trang Admin và User.
* File **bower.json** để chứa các thư viện được include vào project để xử dụng.

# Mô tả các chức năng

## Trang Admin

* Chức năng Login với Admin
* Chức năng Tạo nhóm người dùng
* Chức năng tạo nhóm quyền
* Chức năng chỉnh sửa thông tin user, chỉ admin mới dc chỉnh sửa (Phân quyền)
* Chức năng xem, thêm, xóa, sửa danh mục sản phẩm
* Chức năng xem, thêm, xóa, sửa sản phẩm
* Trang xem thống kê doanh số
* Autocomplete
* Phân trang
* Tạo các Notify khi admin thực hiện thành công 1 hành động

## Trang User

* Đăng nhập với User
* Đăng nhập với Facebook, Google+
* Đăng kí với User
* Xem liên hệ của website
* Xem danh sách sản phẩm theo danh mục
* Xem chi tiết sản phẩm
* Giỏ hàng
* Thanh toán trực tuyến qua Ngân Lượng
* Autocomplete
* Hiển thị Map
* Capcha để chứng thực cho máy móc
* Gửi thông tin mua hàng và xác nhận đăng kí tải khoản qua mail
* Sử dụng Output Cache để load trang nhanh hơn.
* Kĩ thuật SEO cho website

# Hướng dẫn cài đặt

## Yêu cầu hệ thống

* Cài đặt Microsoft SQL Server 2012 trở lên
* Cài đặt Visual Studio 2015 và .NET Framework 4.6.1.
* Có một trong các trình duyệt như Chrome, Cốc cốc, Firefox, vv…
* Có kết nối với mạng Internet.

## Hướng dẫn cài đặt

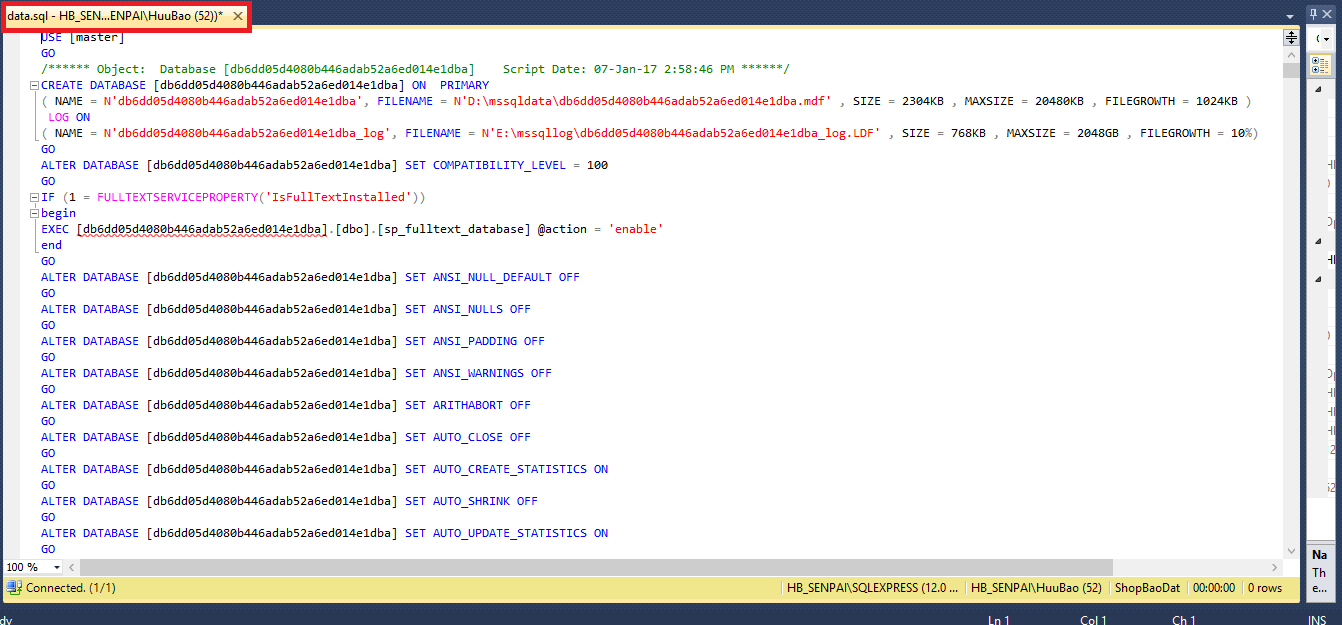
### Cài đặt Database

* Có 2 cách để cài đặt Database

#### Cài đặt bằng cách excute file .sql

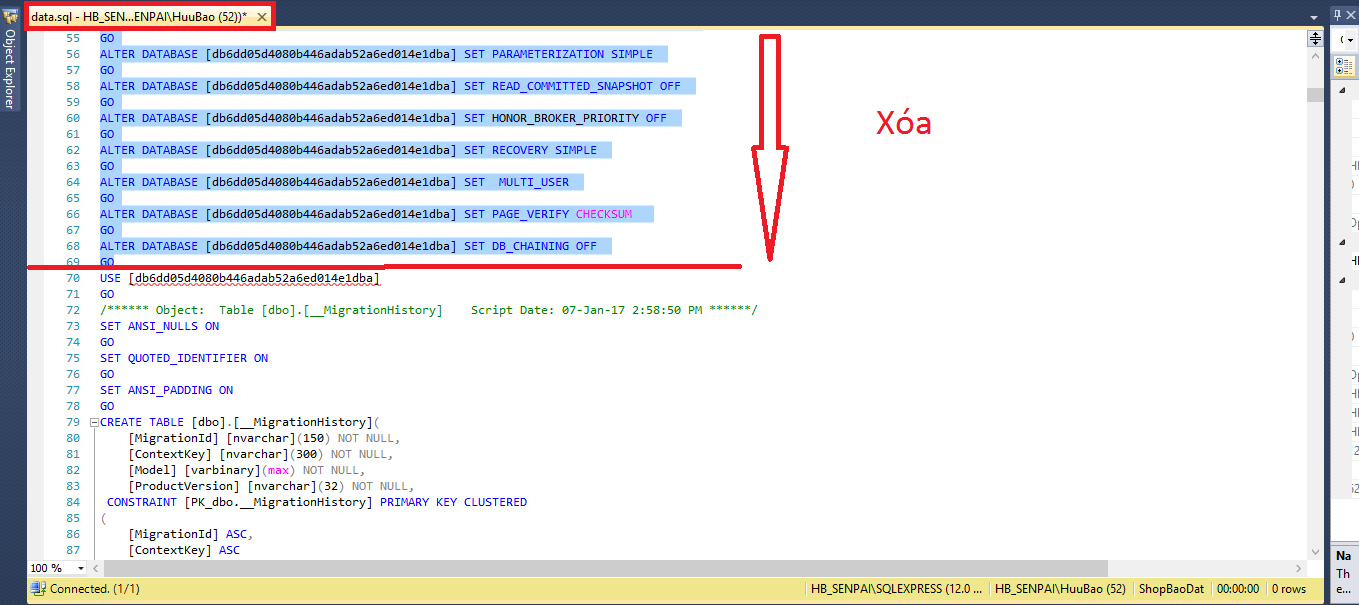
* Cách này sẽ giúp cho CSDL có thể có các dữ liệu mà trang web đang có.

**Bước 1**: Mở file **data.sql**

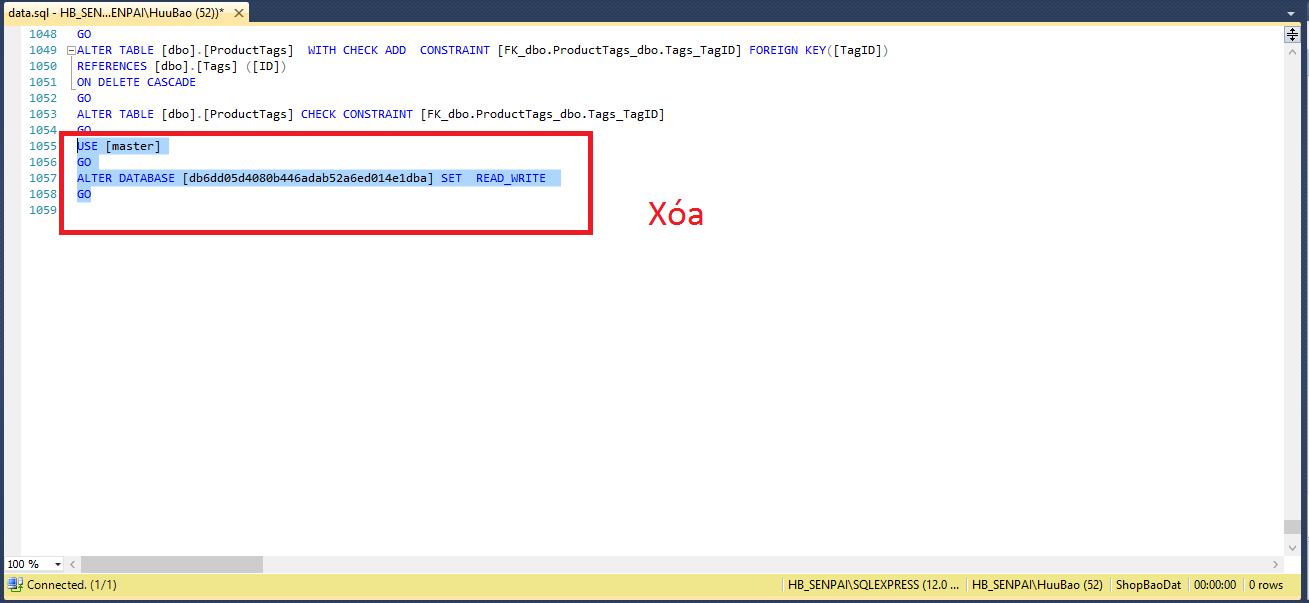


**Bước 2**: Xóa các dòng sau để có thể excute Project:

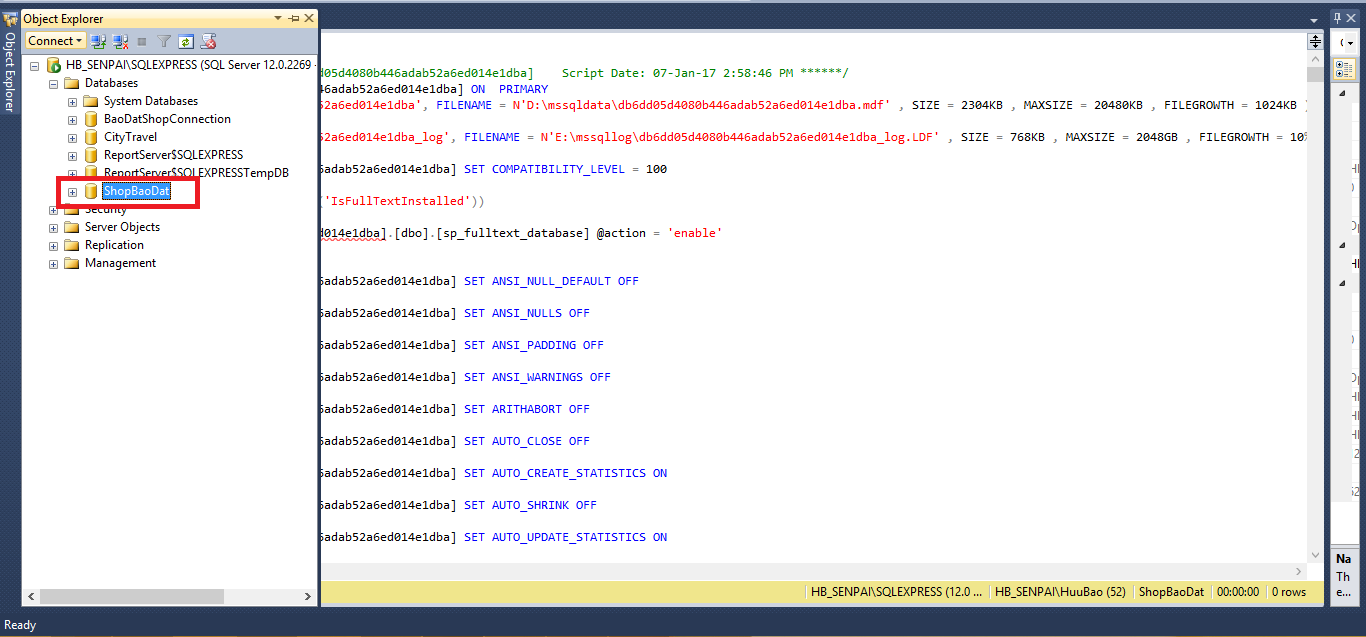
* Dòng đầu tiên đến dòng số 69 trong file **data.sql**, để xem hiển thị số dòng thì chúng ta có thể tham khảo tại link [9] (Hiển thị số dòng trong SQL server 2014 giống với Visual Studio):



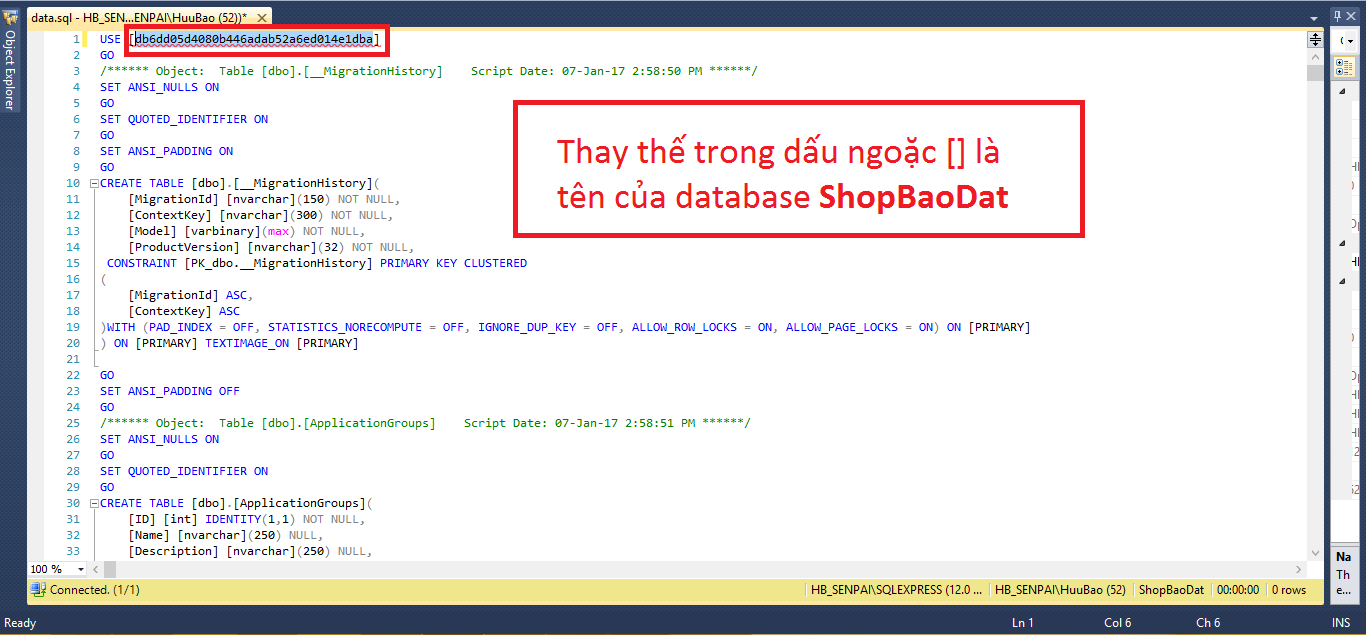
* Và dòng 1055 đến 1059 trong file **data.sql** (4 dòng cuối trong file **data.sql**, vì khi chúng ta xóa những dòng đầu có thể sẽ làm thay đổi số dòng trong file **data.sql**):



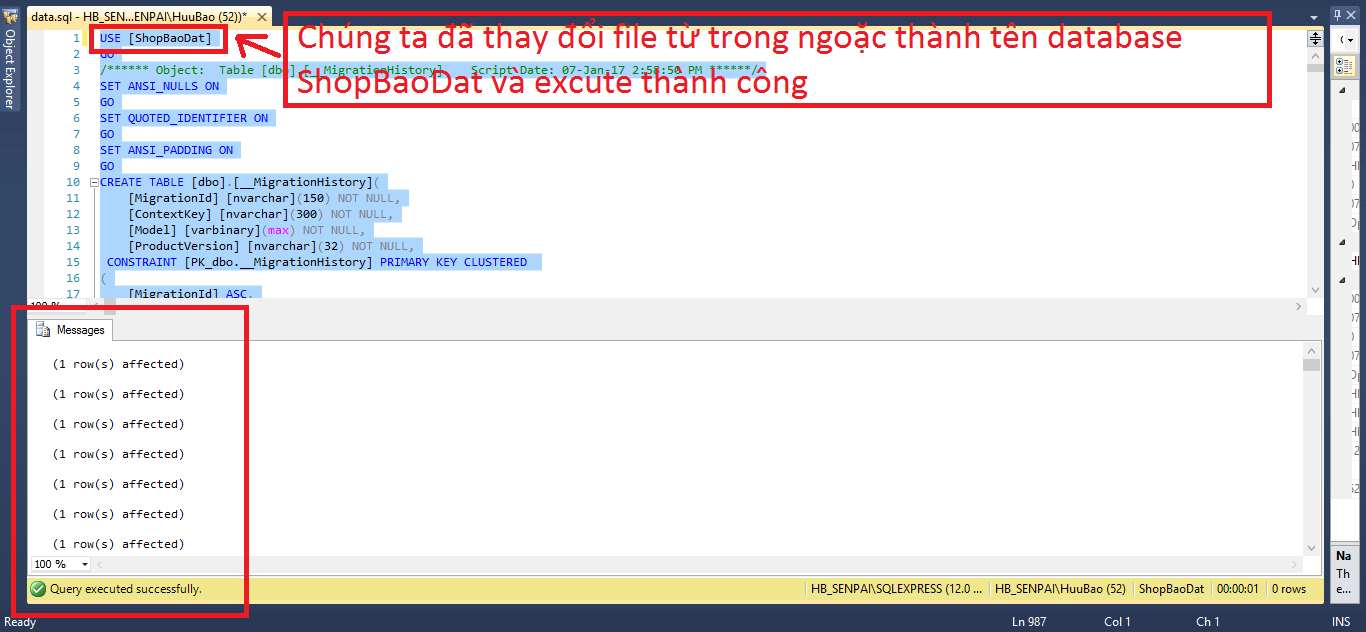
**Bước 3**: Chúng ta tạo 1 database mới có tên là **ShopBaoDat**:



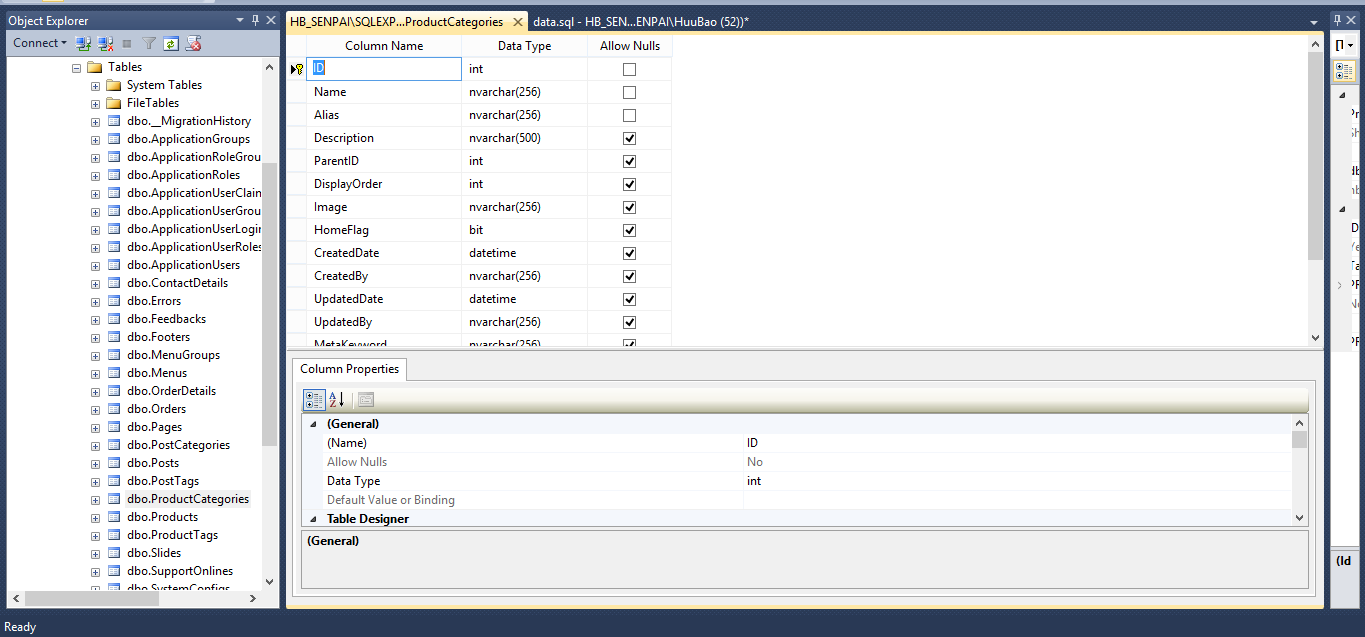
**Bước 4**: Sau đó chúng ta sẽ thay thế tên database vào dòng này:



**Bước 5**: Sau đó nhấn **Excute** để thực thi file **data.sql**:



* Và kết quả là chúng ta có CSDL ShopBaoDat với đầy đủ dữ liệu của trang web <http://shopbaodat.apphb.com/> [10]

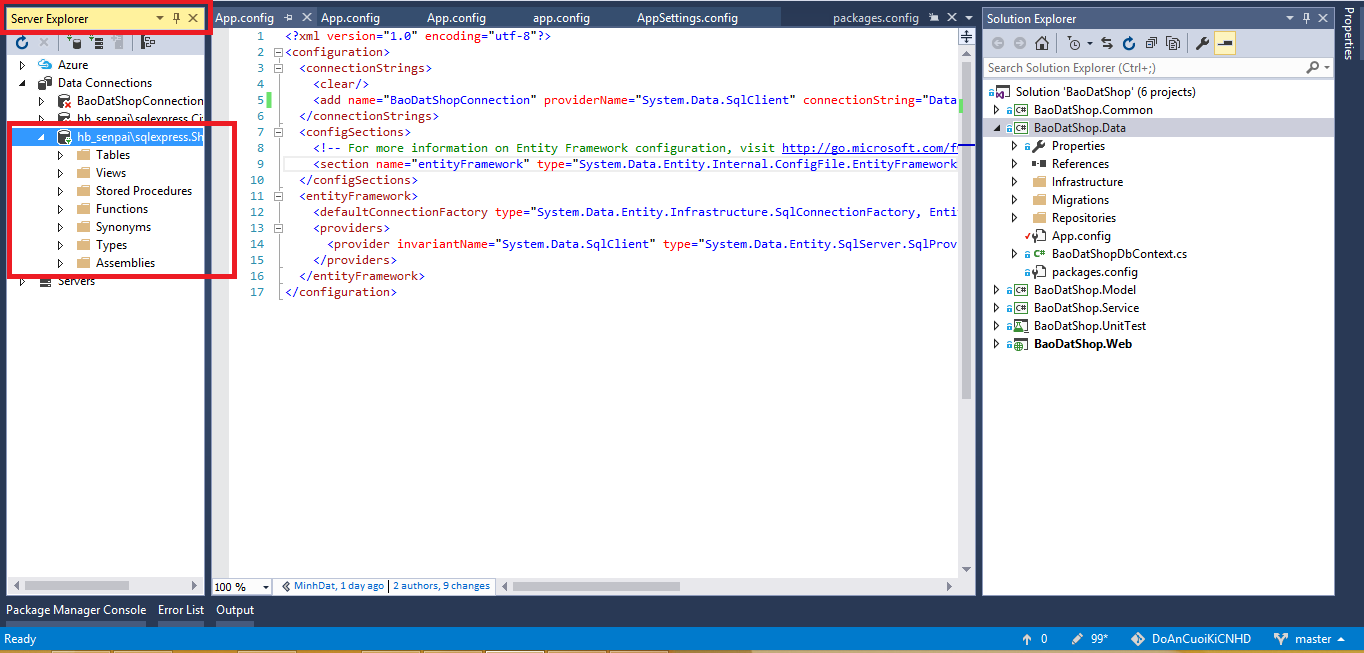


#### Cài đặt bằng cách migrations database trong project Visual Studio 2015:

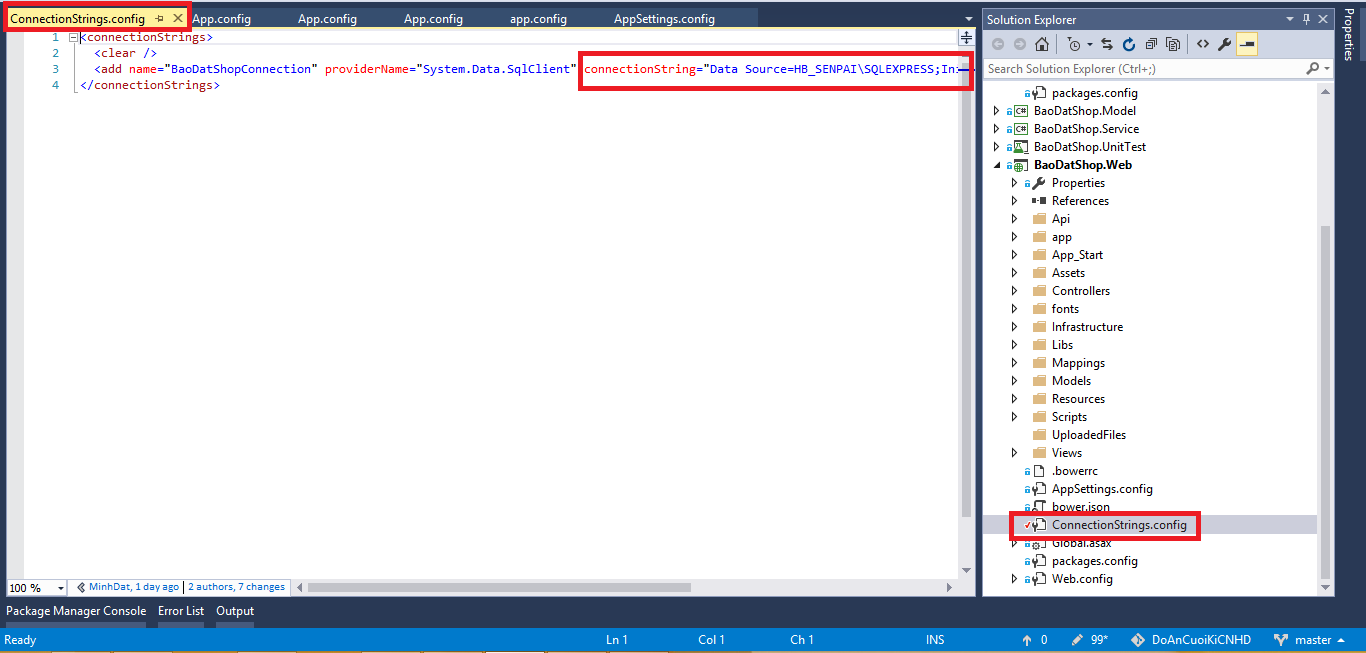
* Cách này sẽ tạo ra được database trong MSSQL server của chúng ta, nhưng sẽ không có được các dữ liệu của website [10]. Vì project được viết theo cơ chế Code First của Entity Framework. Nên Database ta có được chỉ có các bảng, và trong các bảng sẽ không có chưa dữ liệu gì cả.
* Để tạo Database theo cách này ta làm theo các bước sau đây:

**Bước 1:** Chúng ta sẽ tạo ra Database tên ShopBaoDat trong MSSQL của chúng ta (như bước 3 của mục 9.2.1.1)

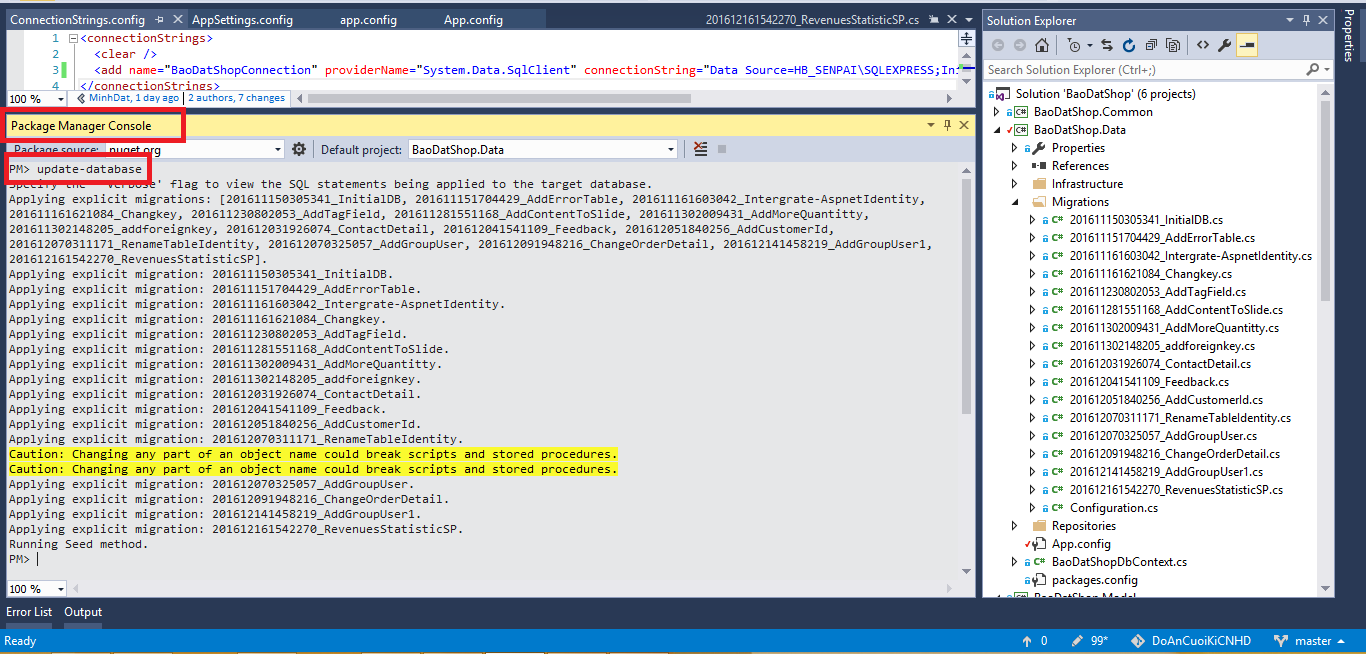
**Bước 2**: Chúng ta sẽ thêm kết nối Data Connection của Database **ShopBaoDat** này với Visual Studio 2015 của chúng ta, làm theo hướng dẫn trong link [11] mục DataConnection để tạo kết nối). Sau khi kết nối ta được:



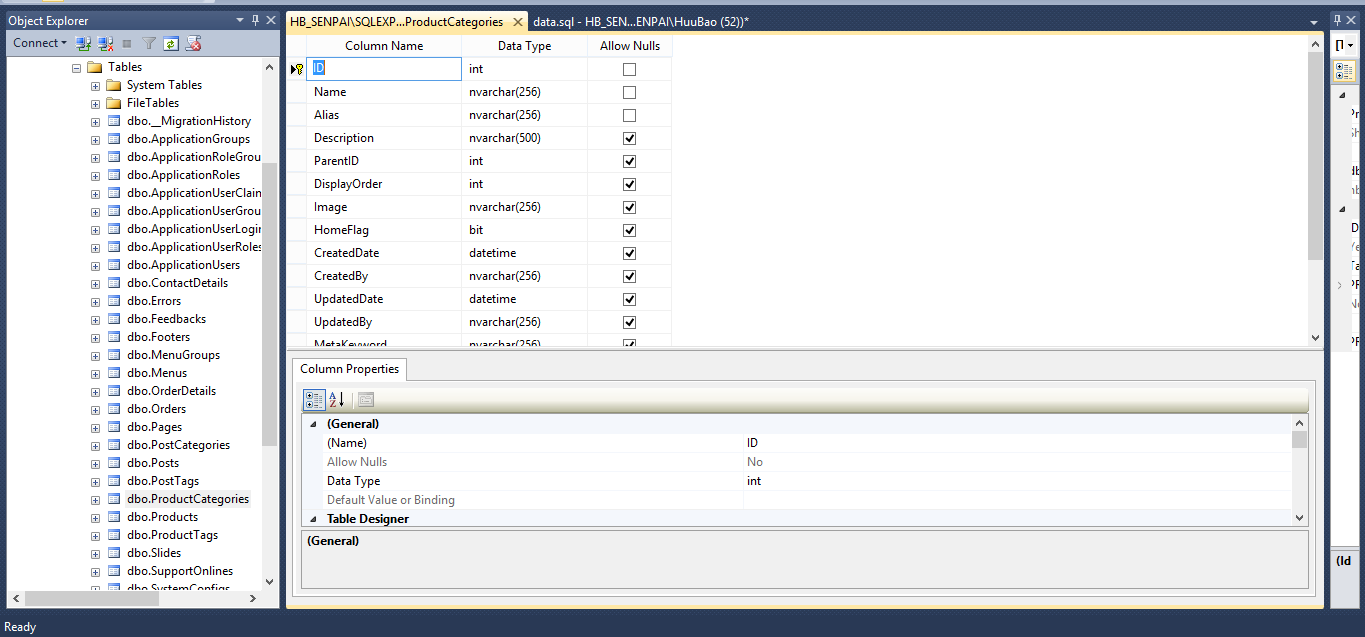
* **Bước 3:** Chỉnh sửa chuỗi kết nối trong 4 file sau của project: **BaoDatShop.Web\ConnectionStrings.config**, **BaoDatShop.UnitTest\app.config**,**BaoDatShop.Data\App.config**, **BaoDatShop.Web\Web.config**



**Bước 4:** Sau khi đổi chuỗi kết nối trong các file xong, thì chúng ta tiến hình Build Project để tiếp theo là bước Migrations Database, bằng cách mở **Package Manager Console** [12] trong Visual Studio. Và chạy câu lệnh **update-database**: Ta được kết quả:

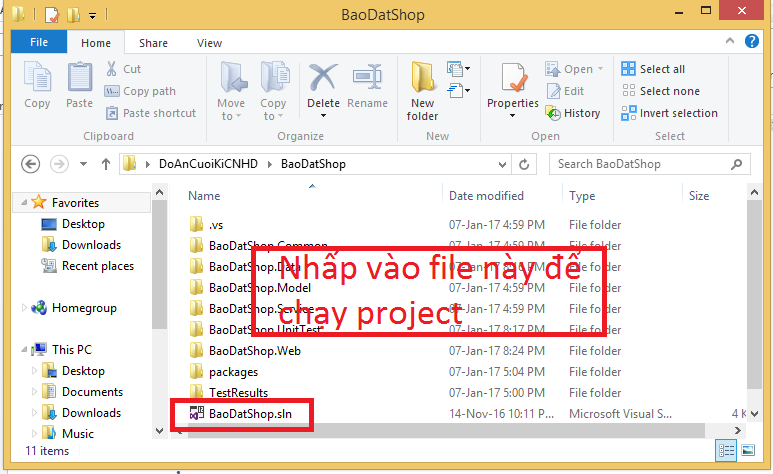


**Bước 5**: Chúng ta được kết quả trong MSSQL của Database **ShopBaoDat** (nhưng database này trong các bảng không có dữ liệu, nên không thể chạy được project nếu chúng ta chưa thêm dữ liệu vào):



### Cài đặt Project trên Visual Studio 2015:

* Vì trong project đã lưu lại các gói package, nên khi mở project lên thì chúng ta chờ cho Visual Studio tải project về. Và tiến hành chạy project kết hợp với Database.



* Trong cách tạo database ở mục **9.2.1.2** thì database chỉ tạo bảng và chưa đổ dữ liệu vào, chúng ta phải tự đổ dữ liệu vào nếu không sẽ bị lỗi không thể chạy được project.

# Hướng dẫn sử dụng

## Trang admin

* Chức năng Login với Admin
* Chức năng Tạo nhóm người dùng
* Chức năng tạo nhóm quyền
* Chức năng chỉnh sửa thông tin user, chỉ admin mới dc chỉnh sửa (Phân quyền)
* Chức năng xem, thêm, xóa, sửa danh mục sản phẩm
* Chức năng xem, thêm, xóa, sửa sản phẩm
* Trang xem thống kê doanh số
* Autocomplete
* Phân trang
* Tạo các Notify khi admin thực hiện thành công 1 hành động

## Trang User

* Đăng nhập với User
* Đăng nhập với Facebook, Google+
* Đăng kí với User
* Xem liên hệ của website
* Xem danh sách sản phẩm theo danh mục
* Xem chi tiết sản phẩm
* Giỏ hàng
* Thanh toán trực tuyến qua Ngân Lượng
* Autocomplete
* Hiển thị Map
* Capcha để chứng thực cho máy móc
* Gửi thông tin mua hàng và xác nhận đăng kí tải khoản qua mail
* Sử dụng Output Cache để load trang nhanh hơn.
* Kĩ thuật SEO cho website

# Kết luận và hướng phát triển

* Source code của website được đặt tên biến khác rõ ràng và phân chia cấu trúc rõ ràng giữa các Client và Server, giữa các lớp với nhau trong mô hình MVC và RestFul Service, để thuận lợi cho việc bảo trì, sửa chữa và phát triển về sau.
* Còn một số tính năng và công cụ mà website chưa thể tích hợp vào như sử dụng framework để xây dựng web mobile, Realtime Push Notification, Sử dụng ESB để giao tiếp giữa các ứng dụng, vv… Tương lai sẽ có thể gắn vào để website có thể đầy đủ hơn.
* Chưa thực hiện tính năng quên mật khẩu, thay đổi mật khẩu trong website.
* Chưa áp dụng được hết các kĩ thuật SEO để có thể góp phần đưa website vào top tìm kiếm từ khóa của Google khi người dùng dùng Google Search để tìm kiếm các website bán hàng trực tuyến qua mạng. Tương lai nhóm sẽ học hỏi đầy đủ các kĩ thuật SEO để áp dụng vào website.

# Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | Microsoft, "https://www.asp.net," [Online]. Available: https://www.asp.net/identity. |
| [2] | Google, "https://developers.google.com," [Online]. Available: https://console.developers.google.com/. |
| [3] | Facebook, "https://developers.facebook.com/," [Online]. Available: https://developers.facebook.com/docs/javascript. |
| [4] | N. Lượng, "https://www.nganluong.vn/," [Online]. Available: https://www.nganluong.vn/nganluong/home.html. |
| [5] | AppHarbor, "https://appharbor.com/," [Online]. Available: https://appharbor.com/. |
| [6] | Cloudinary, "http://cloudinary.com/," [Online]. Available: http://cloudinary.com/. |
| [7] | A. Pierce, "https://almsaeedstudio.com," [Online]. Available: https://almsaeedstudio.com/themes/AdminLTE/index2.html. |
| [8] | w3layouts, "https://w3layouts.com/," [Online]. Available: https://w3layouts.com/big-shope-flat-ecommerce-bootstrap-responsive-web-template/. |
| [9] | TEK, "http://tek.eten.vn," [Online]. Available: http://tek.eten.vn/hien-thi-dong-line-number-trong-visual-studio. |
| [10] | B. v. Dat, "http://shopbaodat.apphb.com/," [Online]. Available: http://shopbaodat.apphb.com/. |
| [11] | o7planning, "http://o7planning.org/," [Online]. Available: http://o7planning.org/vi/10511/ket-noi-database-sql-server-su-dung-csharp. |
| [12] | http://www.c-sharpcorner.com/, "Ketak Bhalsing," [Online]. Available: http://www.c-sharpcorner.com/UploadFile/22da8c/package-manager-console-in-visual-studio/. |