**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**BÁO CÁO ĐỒ ÁN MÔN HỌC CÔNG NGHỆ .NET**

**ĐỀ TÀI**

**ASP.NET MVC**

Giảng viên hướng dẫn: ThS Phạm Thi Vương

Nhóm thực hiện:

Phạm Tuấn Tài 15520749

Nguyễn Thanh Huy 15520310

Nguyễn Tấn Phú 15520624

Nguyễn Bảo Hùng 15520277

Lời cảm ơn

Chúng em xin chân thành cảm thầy Phạm Thi Vương giảng viên khoa công nghệ phần mềm, trường đại học Công nghệ thông tin – Đại học quốc gia TP.HCM trong thời gian qua đã giảng dạy, hỗ trợ nhóm chúng em để thực hiện đồ án này. Nhóm đã cố gắng vận dụng hết kiến thức của mình để hoàn thành đồ án một cách tốt nhất. Tuy nhiên, trong quá trình thực hiện khó tránh khỏi sai sót vì thế nhóm chúng em rất mong nhận được những ý kiến đóng góp của thầy để có thể hoàn thành đồ án tốt hơn.

Một lần nữa chúng em xin chân thành cảm ơn thầy.

*Nhóm thực hiện.*

Phạm Tuấn Tài – 15520749

Nguyễn Thanh Huy - 15520310

Nguyễn Tấn Phú - 15520624

Nguyễn Bảo Hùng - 15520277

Mục lục

[CHƯƠNG 1: GIỚI THIỆU VỀ ASP.NET MVC 4](#_Toc515768121)

[1.1 Mô hình MVC là gì? 4](#_Toc515768122)

[1.1.1 Các thành phần trong mô hình MVC: 4](#_Toc515768123)

[1.2 ASP.NET là gì? 4](#_Toc515768124)

[1.3 ASP.NET MVC là gì? 5](#_Toc515768125)

[1.4 Tại sao nên sử dụng ASP.NET MVC? 5](#_Toc515768126)

[1.4.1 Phân tách các mối bận tâm (concern): 5](#_Toc515768127)

[1.4.2 Controls đa dạng: 5](#_Toc515768128)

[1.4.3 Có khả năng kiểm thử: 5](#_Toc515768129)

[1.4.4 Gọn nhẹ: 5](#_Toc515768130)

[1.4.5 Tích hợp: 5](#_Toc515768131)

[CHƯƠNG 2: CÁC THAO TÁC CƠ BẢN VỚI ASP.NET MVC 6](#_Toc515768132)

[2.1 Tạo ứng dụng đầu tiên: 6](#_Toc515768133)

[2.2 Cấu trúc của một project ASP.NET MVC: 9](#_Toc515768134)

[2.3 Tạo controller: 10](#_Toc515768135)

[2.4 Truyền parameter ở URL: 13](#_Toc515768136)

[2.5 Tạo View: 14](#_Toc515768137)

[2.6 Dùng ViewBag để chuyển dữ liệu từ controller sang view 15](#_Toc515768138)

[2.7 Tạo model: 16](#_Toc515768139)

[CHƯƠNG 3: KẾT NỐI VỚI CƠ SỞ DỮ LIỆU 20](#_Toc515768140)

[3.1 Chuẩn bị cơ sở dữ liệu 20](#_Toc515768141)

[3.2 Kết nối cơ sở dữ liệu: 21](#_Toc515768142)

[CHƯƠNG 4: LAYOUT VÀ UI 30](#_Toc515768143)

[4.1 Nhúng CSS vào trang web: 30](#_Toc515768144)

[4.1.1 CSS là gì 30](#_Toc515768145)

[4.1.2 Sử dụng css: 30](#_Toc515768146)

[4.2 Reponsive design với boostrap 38](#_Toc515768147)

[4.2.1 Boostrap la gì? 38](#_Toc515768148)

[4.2.2 Sử dụng boostrap 3 38](#_Toc515768149)

[CHƯƠNG 5: SESSION 40](#_Toc515768150)

[5.1 Cách sử dụng session: 40](#_Toc515768151)

[CHƯƠNG 6: FACEBOOK COMMENT PLUGIN 41](#_Toc515768152)

[6.1 Tại sao nên sử dụng Facebook comment plugin: 41](#_Toc515768153)

[6.2 Cách nhúng Facebook comment plugin vào website 41](#_Toc515768154)

[CHƯƠNG 7: Các thành phần khác 45](#_Toc515768155)

[7.1 Ajax 45](#_Toc515768156)

[7.1.1 Ajax là gì: 45](#_Toc515768157)

[7.1.2 Sử dụng Ajax: 45](#_Toc515768158)

[7.2 jQuery: 46](#_Toc515768159)

[7.2.1 jQuere là gì: 46](#_Toc515768160)

[7.2.2 Sử dụng jQuery 46](#_Toc515768161)

[7.3 Google Maps: 49](#_Toc515768162)

[7.3.1 Google maps là gì 49](#_Toc515768163)

[7.3.2 Map Api 49](#_Toc515768164)

[7.3.3 Sử dụng 49](#_Toc515768165)

[CHƯƠNG 8: KẾT LUẬN 50](#_Toc515768166)

# GIỚI THIỆU VỀ ASP.NET MVC

## Mô hình MVC là gì?

Mô hình MVC (Model – View – Controller) là một kiến trúc phần mềm hay mẫu thiết kế được sử dụng trong kỹ thuật phần mềm. Nó giúp cho các lập trình viên tách ứng dụng của họ ra 3 thành phần khác nhau Model, View và Controller. Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác



### Các thành phần trong mô hình MVC:

* **Model:** Chứa các phương thức xử lý, truy xuất cơ sở dữ liệu, các đối tượng mô tả dữ liệu như các class.
* **View:** Tiếp nhận dữ liệu từ controller và hiển thị cho người dùng. View cũng là nới chứa các thành phần giao diện để người dùng tương tác với hệ thống.
* **Controller:** Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng. Controller tiếp nhận yêu cầu từ client sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client.

## ASP.NET là gì?

ASP.NET là một web application framework 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 web service.

ASP.NET được biên dịch dưới dạng Common Language Runtime (CLR), cho phép những người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi .NET language.

## ASP.NET MVC là gì?

ASP.NET MVC là một framework web được phát triển bởi Microsoft, áp dụng mô hình MVC (Model–View–Controller). ASP.NET MVC là framework mã nguồn mở, tách rời với thành phần độc quyền ASP.NET Web Forms. ASP.NET MVC đang nổi lên là phương pháp phát triển web mạnh nhất và phổ biến nhất trên nền ASP.NET hiện nay.

## Tại sao nên sử dụng ASP.NET MVC?

### Phân tách các mối bận tâm (concern):

Framework này cung cấp sự phân tách rất rõ gàng giữa model, logic nghiệp vụ, giao diện người dùng và dữ liệu.

### Controls đa dạng:

Được cung cấp một bộ control rất mạnh mẽ trên JavaScript, HTML và CSS so với các control được cung cấp bởi một số hình thức truyền thống trên web.

### Có khả năng kiểm thử:

MVC framework cho phép kiểm thử mà không cần triền khai trên một web server (IIS). Visual Studio tích hợp công cụ kiểm thử mã nguồn mỡ Nunit giúp việc kiểm thử trên các ứng dụng web ASP.NET MVC dễ dàng và thuận tiện hơn.

### Gọn nhẹ:

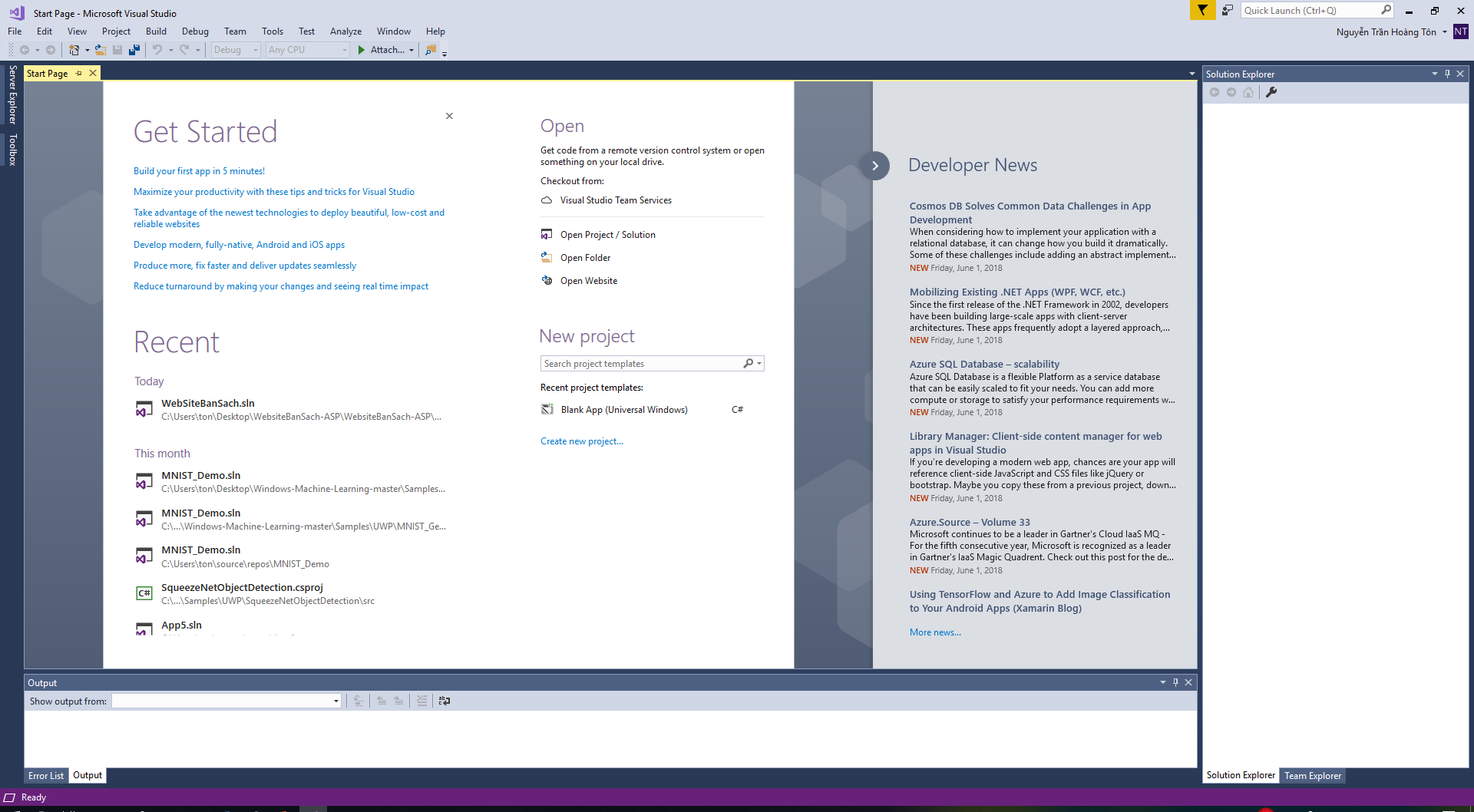
MVC không có một ViewState. Nó sử dụng model/ViewModels để chuyển dữ liệu qua lại giữa các Views. Không sử dụng View State, hỗ trợ bạn trong việc làm giảm băng thông của các request rất nhiều.

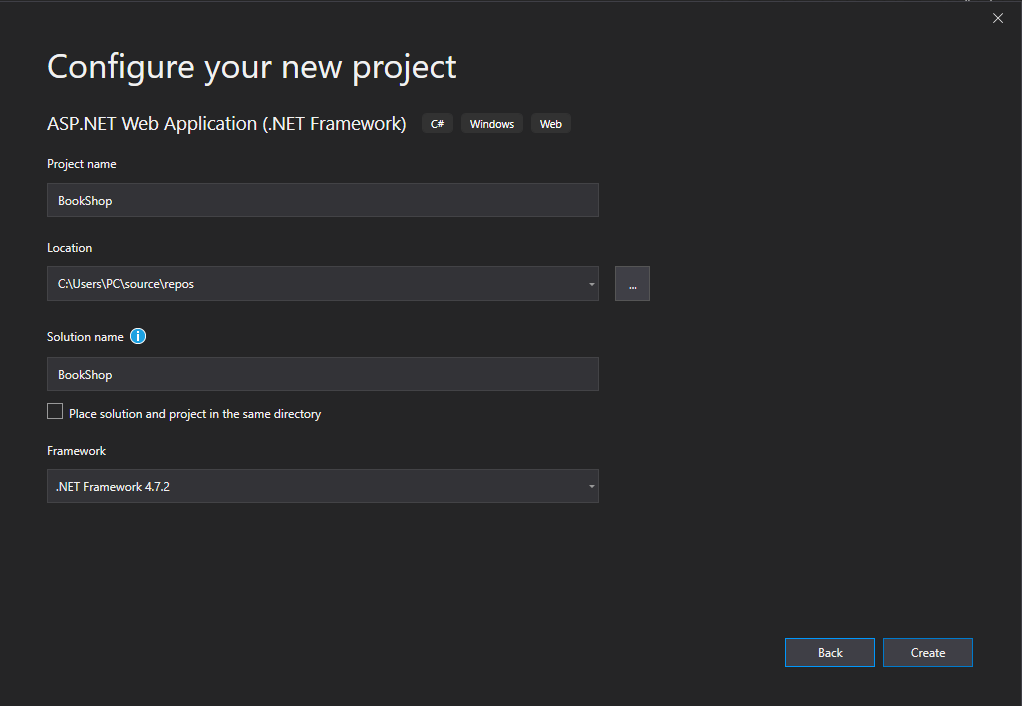
### Tích hợp:

Có thể tích hợp jQuery để xử lý ngay tại trình duyệt. Giảm tải cho server.

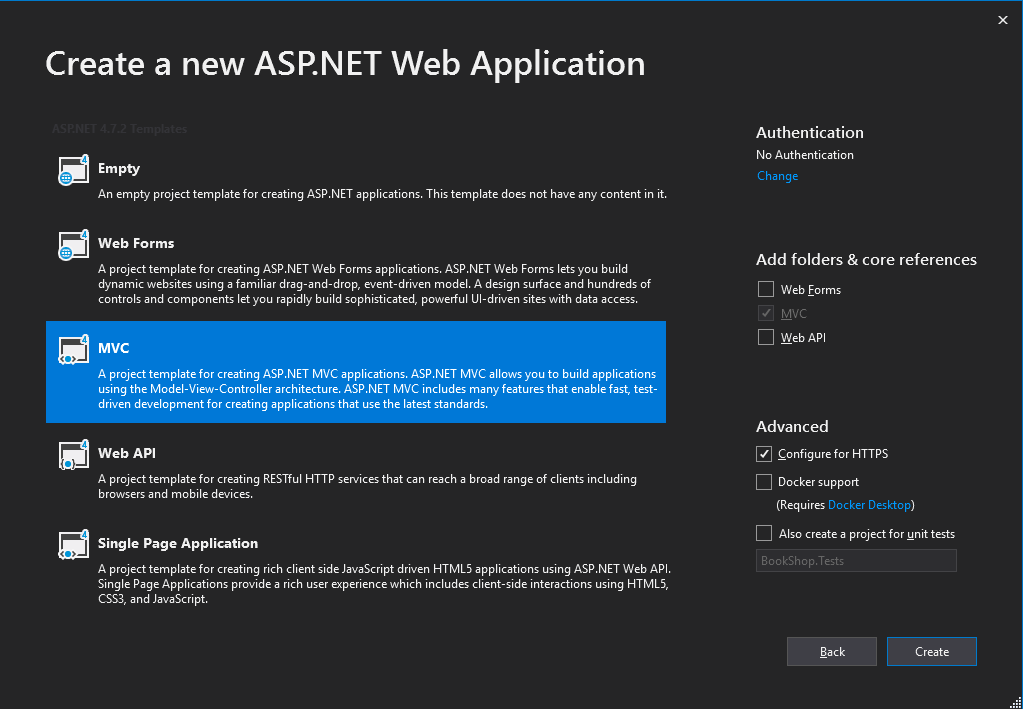
# CÁC THAO TÁC CƠ BẢN VỚI ASP.NET MVC

## Tạo ứng dụng đầu tiên:

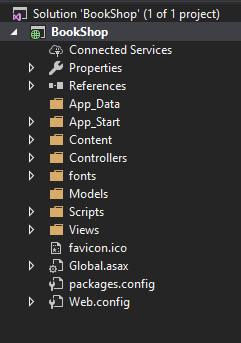
1. Mở Visual Studio
2. Chọn **File > New Project**, tại cửa sổ mới hiện ra chọn **Visual C# > Web > ASP.NET Web Application.** Đặt tên cho project rồi chọn OK.

****

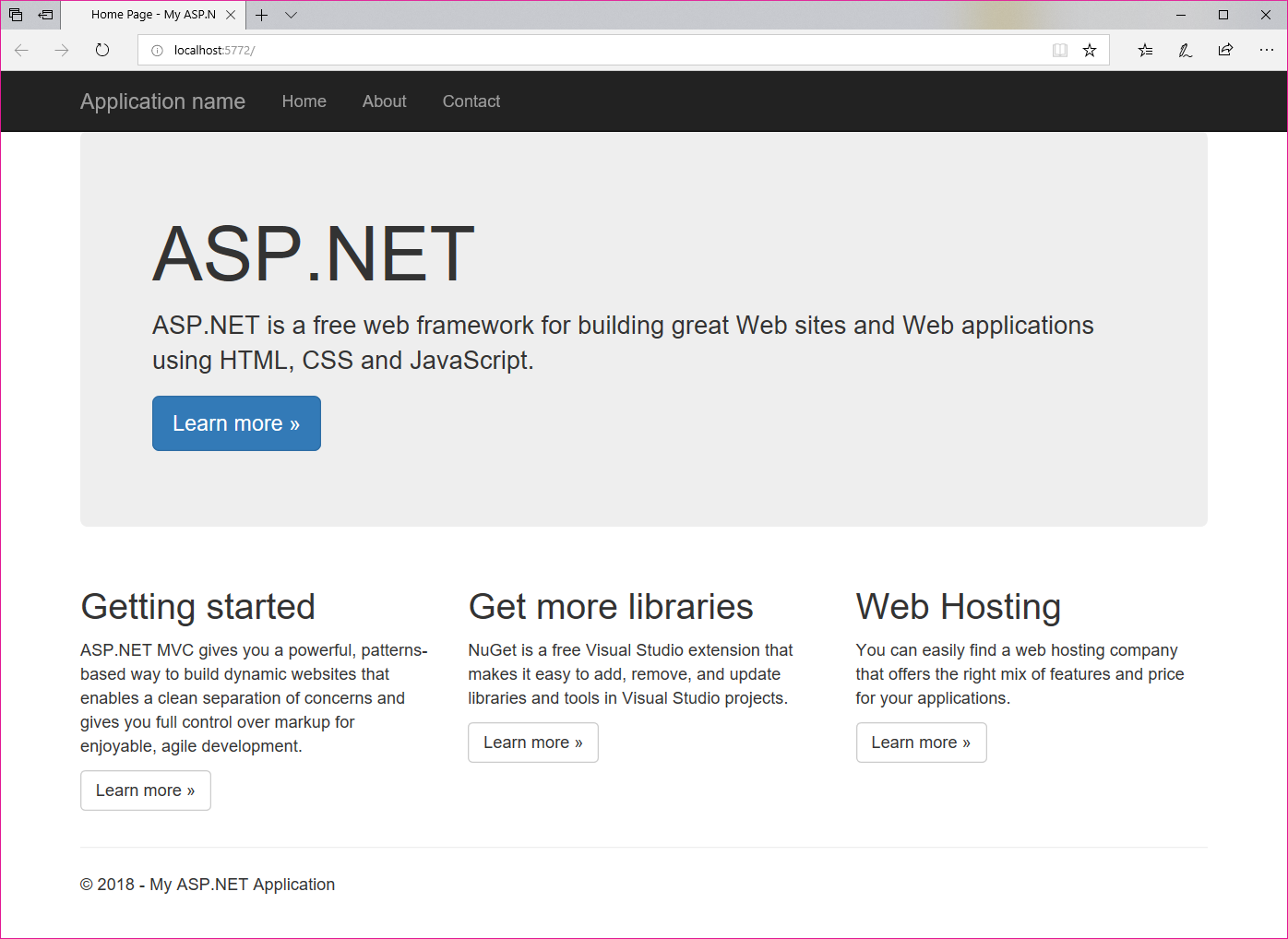
1. Tại cửa sổ mới Chọn **MVC** rồi chọn **OK.**



1. Đợi một lúc Visual Studio sẽ tạo cho bạn một project với cấu trúc sau.

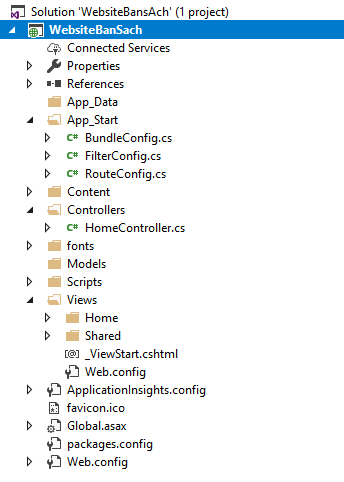


1. Bấm **F5** để chạy thử project đầu tiên.



## Cấu trúc của một project ASP.NET MVC:

Xem kĩ hơn cấu trúc thư mục của project ta vừa tạo ở mục **2.1:**



**Properties:** Cái này chứa các thuộc tính của project. Chúng ta tạm thời không quan tâm đến file này.

**References:** Cái này chứa các thư viện, package mà chúng ta sử dụng. Tương tự như vendor trong PHP hay node-modules trong node js

**App\_Data:** Thư mục chứa file dữ liệu của hệ thống. Cái này cũng tạm thời chưa cần quan tâm.

**App\_Start:** Thư mục chưa các file cấu hình khởi động và biên dịch của project. Chúng ta chú ý 2 file đó là FilterConfig.cs, dùng để khai báo các filter sử dụng trước khi thực hiện 1 hành động nào đó và file RouteConfig.cs để chứa các cài đặt đều hướng trang.

**Content:** Thư mục chứa CSS.

**Controllers:** chứa các controller.

**Models:** Chứa các file tương tác với CSDL.

**Scripts:** Thư mục chứa các file JS.

**Views:** Thư mục chứa các file HTML với đuôi là .cshtml.

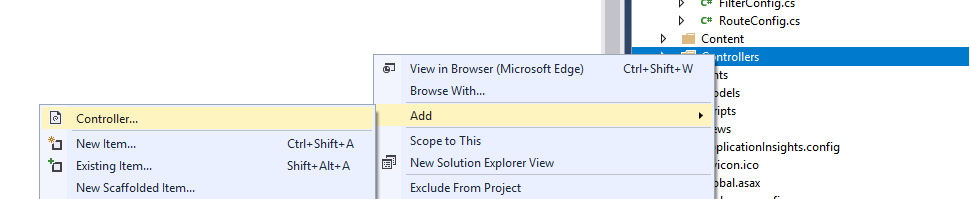
**Global.asax:** File chứa các khai báo sử dụng cho toàn bộ project, như định dạng số, ngày tháng, vùng, ngôn ngữ.

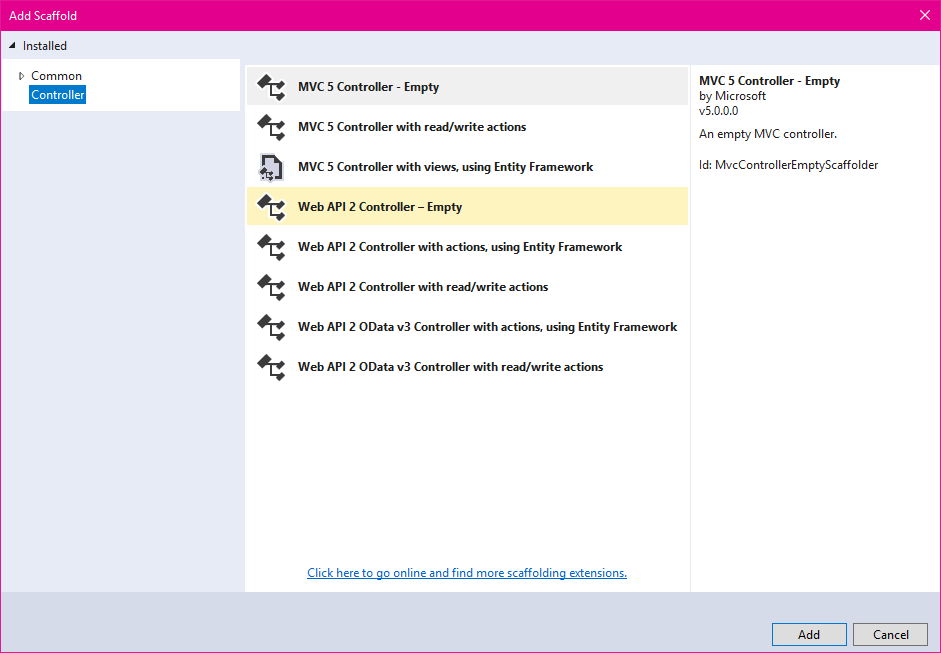
**packages.config:** File quản lý các package chúng ta cài vào

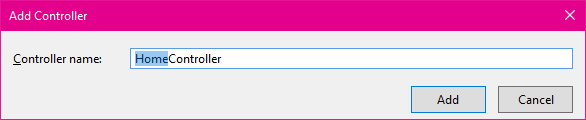
**Web.config:** File này khá quan trọng, nó định nghĩa tất cả cài đặt cho project.

## Tạo controller:

1. Chuột phải vào thư mục **Controllers**, chọn **Add > Controller**



1. Chọn MVC5 Controller -Empty và chọn Add: 
2. Đặt tên controller rồi chọn Add



Lưu ý: Khi tạo project mặc định Visual Studio đã tạo sẵn HomeController nên sẽ xuất hiện hộp thoại hỏi có đồng ý thay thế file cũ hay không, ta chọn yes.

Ta được file như sau:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace BookShop.Controllers {

public class HomeController : Controller

{

// GET: Home

public ActionResult Index()

{

return View();

}

}

}

Trong class HomeController có một phương thức Index kiểu trả về ActionResult. Phương thức này sẽ trả về một file cshtml trùng tên. Để trả về 1 trang html có tên khác với tên Action đang thực thi thì chỉ cần truyền tên của file chứa trang html đó như sau return View("tenFile");

1. Ta thay đổi các phương thức trong HomeController như sau:

namespace BookShop.Controllers {

{

public class HomeController : Controller

{

// GET: Home

public String Index()

{

return "Đây là phương thức index mặc định của controller home";

}

public String Hello()

{

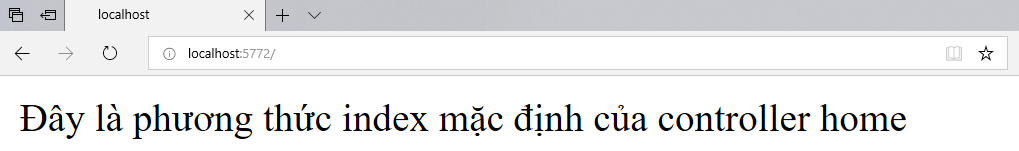
return "xin chào, đây là phương thức Hello trong controller home";

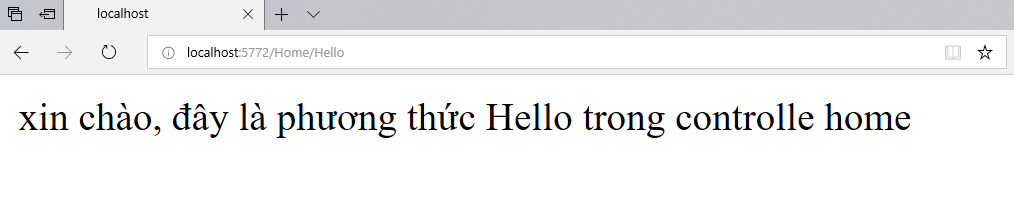
}

}

}

1. Bấm F5 để chạy ứng dụng, ở trình duyệt thử chạy 2 địa chỉ [**http://localhost:5772/**](http://localhost:5772/) **và** [**http://localhost:5772/Home/Hello**](http://localhost:5772/Home/Hello)





ASP.NET MVC sẽ gọi các lớp controller khác nhau (kèm theo các phương thức) tùy theo URL đầu vào. Mặc định URL có định dạng:

…/[Controller]/[ActionName]/[Parameters]

Phần … là địa chỉ website của bạn, ví dụ http://www.google.com/ hay http://localhost:xxxx/. Controller là tên phần đầu của Controller, ví dụ HomeController.cs thì có tên là Dammio. ActionName là tên phương thức, ví dụ phương thức Hello. Parameters là các tham số đầu vào của các phương thức. Vì trong ví dụ trên, chúng ta chưa tạo Parameters cho phương thức vì vậy phần này trống.

Cấu trúc đường dẫn được quy định tại **App\_Start/RouteConfig.cs**

namespace BookShop.Controllers {

{

public class RouteConfig

{

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }

);

}

}

}

Có thể thấy giá trị mặc định của đường dẫn controller là Home, actionn là Index vì thế nên khi ta vào địa chỉ <http://localhost:5772/> sẽ gọi đến phương thức Index trong controller Home.

## Truyền parameter ở URL:

1. Để truyền parameters trên URL ta thay đổi phương thức Hello như sau:

public String Hello( String name = 'bạn")

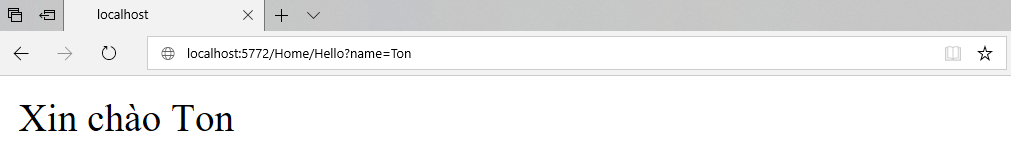
{

return HttpUtility.HtmlEncode("Xin chào " + name);

}

* + 1. **Chú ý:** Bạn sử dụng HttpUtility.HtmlEncode để chống người dùng chèn script độc hại vào chương trình. Ví dụ như:
    2. **/Store/Browse?Genre=<script>window.location=’http://hackersite.com’</script>**.

1. Bấm F5 hoặc Ctrl-F5 để chạy ứng dụng và nhập URL:

[**http://localhost:5772/Home/Hello?name=Ton**](http://localhost:5772/Home/Hello?name=Ton)

## Tạo View:

Khi tạo một controller Visual Studio sẽ tự thêm vào thư mục Views một thư mục trùng tên với controller vừa tạo. thư mục này dùng để chứa các trang cshtml của controller tương ứng.

1. Sửa phương thức Index trong Controllers/HomeController như sau

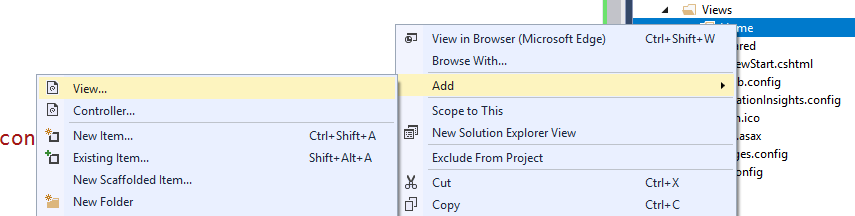
public ActionResult Index()

{

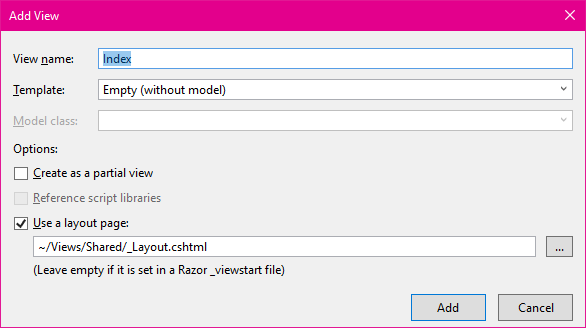
return View();

}

1. Click chuột phải vào thư mục **/Views/Home** và **chọn Add > View**



1. Đặt tên cho View là Index (View phải cùng tên với action) và tại mục Use a layout page chọn đường dẫn /View/Shared/\_Layout.cshtml (đây là file layout mặc định)



1. Visual Studio sẽ tạo một file Index.cshtml như sau:

@{

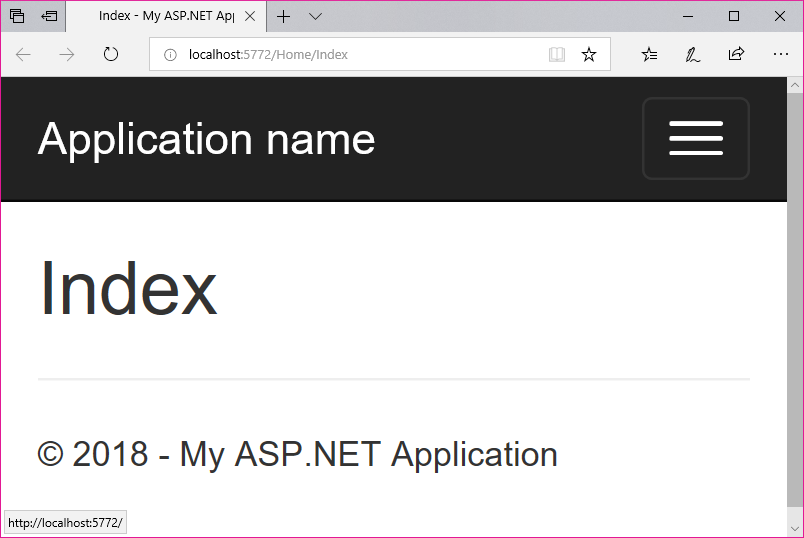
ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Index</h2>

1. Click chuột phải vào file Index.cshtml > View in browser



## Dùng ViewBag để chuyển dữ liệu từ controller sang view

1. Ta thay đổi phương thức Index tại /Controller/HomeController như sau:

public ActionResult Index()

{

ViewBag.Books = new List<string> { "Conan", "Doraemon", "Pokemon", "Dragon Balls", "One Piece" };

return View();

}

1. Ta thay đổi nội dung /View/Home/Index.cshtml như sau:

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h3>List books</h3>

<p>List books from ViewBag</p>

<ul>

@foreach (string book in ViewBag.Books)

{

<li>

@book

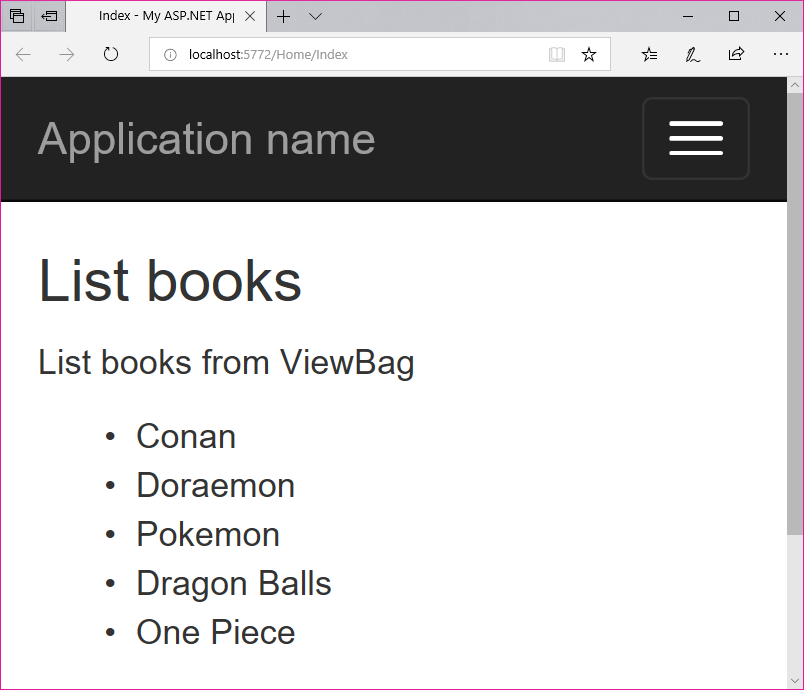
</li>

}

</ul>

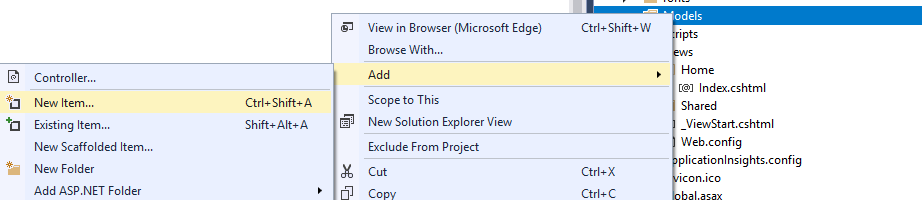
Trong file .cshtml các câu lệnh C# sẽ đặt sau kí tự @.

1. Nhân F5 và chờ xem kết quả

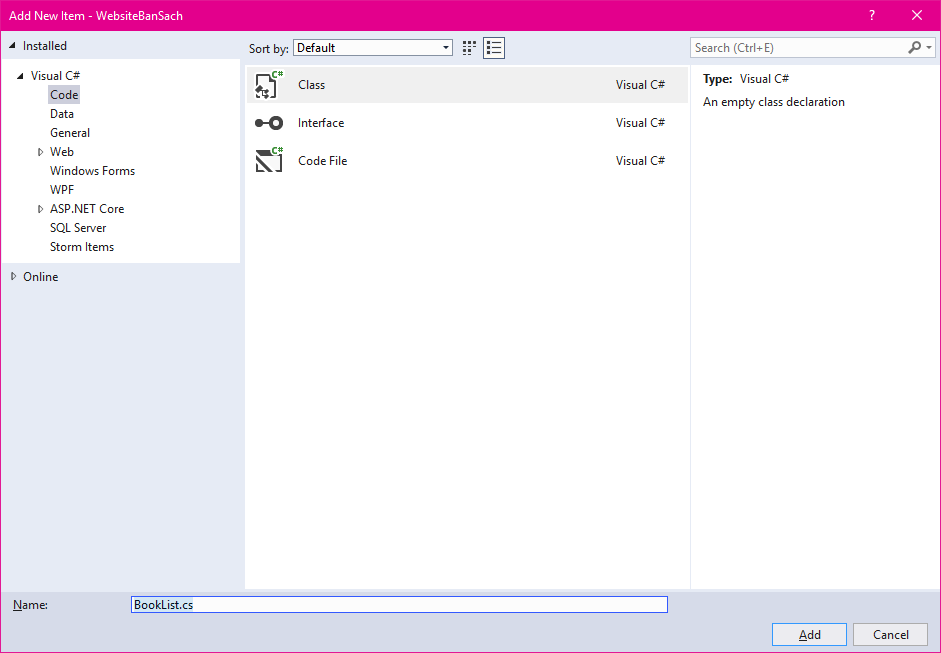


## Tạo model:

1. Click chuột phải vào thư mục Models chọn Add > New item



1. Tại cửa số mới hiện ra ở menu bên phải chọn **Visual C# > Code > Class** nhập tên class và chọn **Add**



1. Thay đổi file BookList.cs vừa tạo như sau:

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace BookShop.Controllers {

{

public class BookList

{

public List<string> Genres { get; set; }

}

}

1. Thay đổi HomeController.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using WebsiteBansAch.Models;

namespace WebsiteBansAch.Controllers

{

public class HomeController : Controller

{

// GET: Home

public ActionResult Index()

{

var books = new List<string> { "Conan", "Doraemon", "Pokemon", "Dragon Balls", "One Piece" };

//create model

var booksModel = new BookList

{

list = books

};

return View(booksModel);

}

}

}

1. Thay đổi /Views/Home/Index.cshtml:

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

@model WebsiteBansAch.Models.BookList

<h3>List books</h3>

<p>List books from Model</p>

<ul>

@foreach (string book in Model.list)

{

<li>

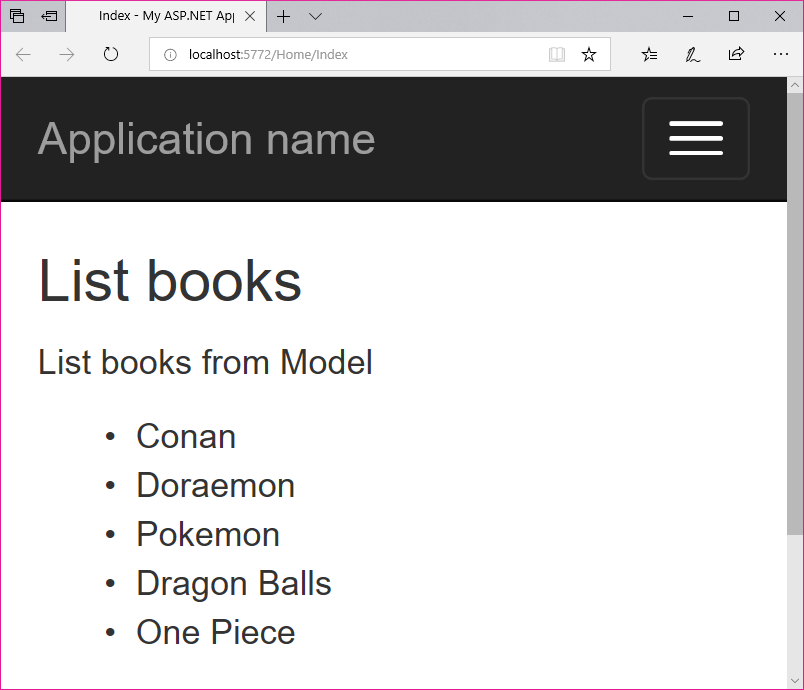
@book

</li>

}

</ul>

1. Nhấn F5 để chạy ứng dụng

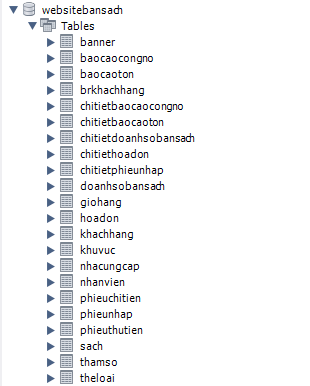


Vậy là ta đã chuyển thành công dữ liệu từ Models đến View.

# KẾT NỐI VỚI CƠ SỞ DỮ LIỆU

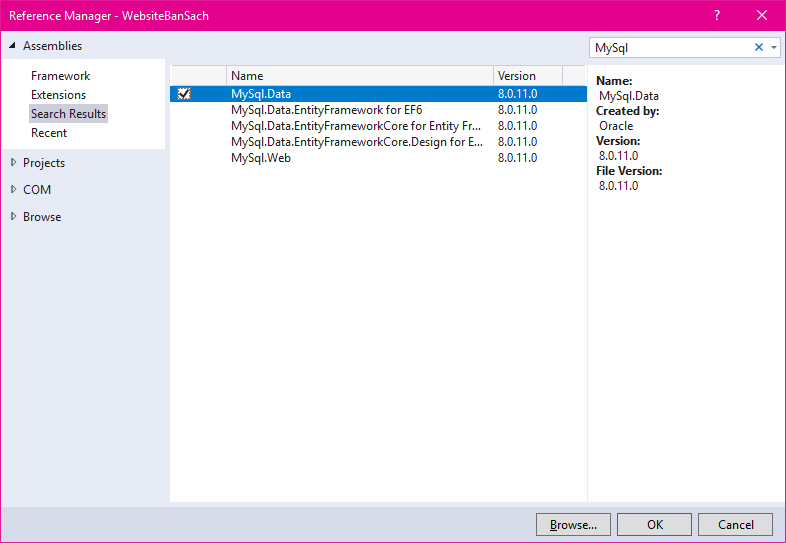
## Chuẩn bị cơ sở dữ liệu

1. Tải MySQL server tại link <https://dev.mysql.com/downloads/> và tiến hành cài đặt
2. Tải xuống file [SQL](https://drive.google.com/file/d/1uDV0ieSmea5uJB5OJdezPn2kojmL_tV3/view?usp=sharing)
3. Import file tải được vào MySQL server ta được kết quả như sau:

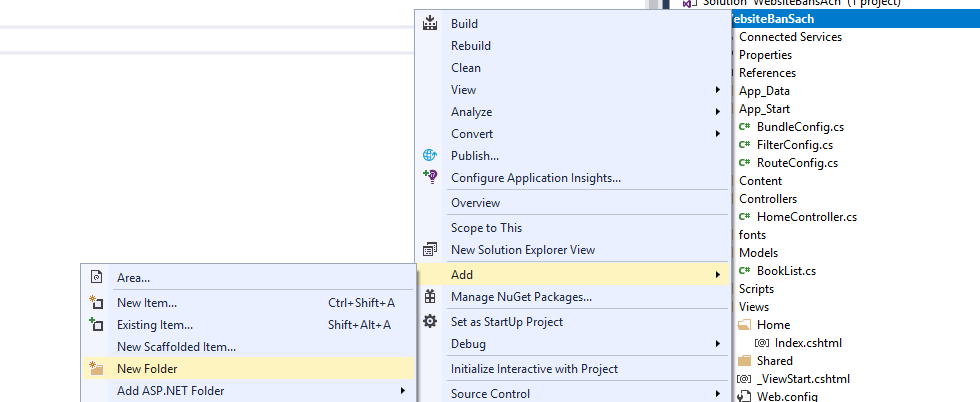


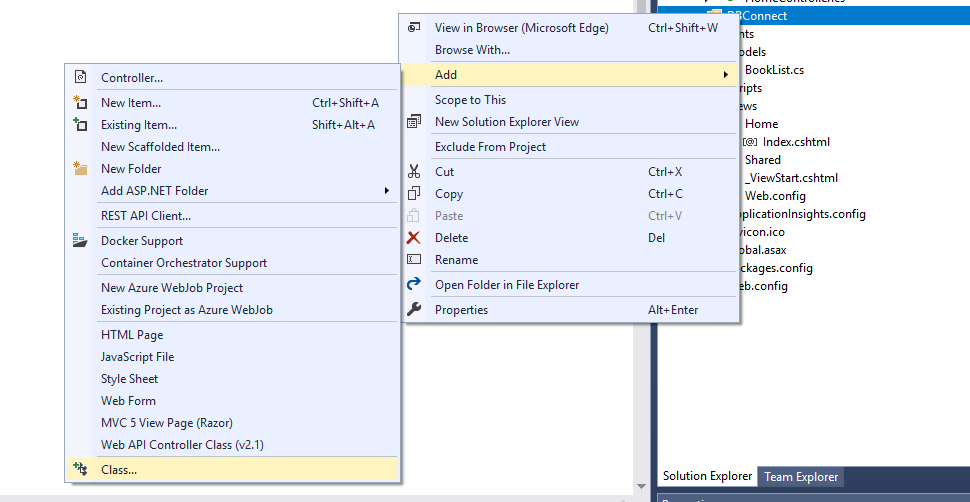
## Kết nối cơ sở dữ liệu:

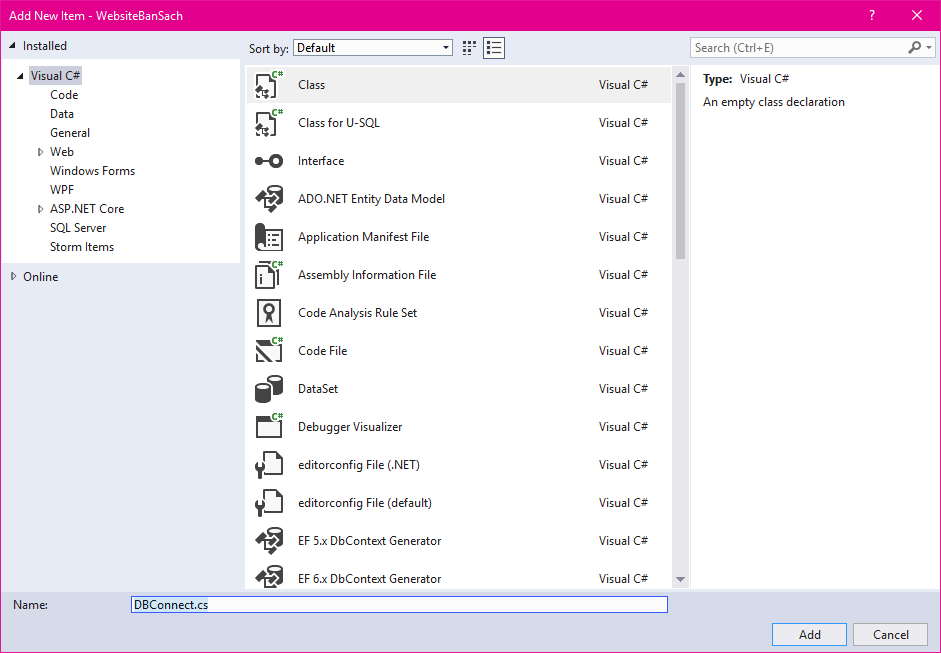
1. Tải xuống MySQL Connector tại <https://dev.mysql.com/downloads/connector/net/>
2. Sau khi tải về tiến hành cài đặt MySQL Connector
3. Trở lại project, ở solution explorer click chuột phải vào References chọn Add reference
4. Ở thanh tìm kiếm góc trên bên phải của cửa sổ mới hiện ra nhập vào “MySql”
5. Tick chọn MySql.Data rồi chọn OK.



1. Click chuột phải vào Project tại Solution explorer chọn Add > New Folder đặt tên là Connect.



1. Click chuột phải vào thư mục vừa tạo chọn Add > Class và đặt tên là DBConnect



1. Thêm vào file vừa tạo nội dung sau:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using MySql.Data.MySqlClient;

namespace WebSiteBanSach.Connect

{

public class DBConnect

{

public static MySqlConnection getConnection()

{ MySqlConnection connect = null;

//Lưu ý nhập đúng tên use, password và database của MySQL

connect = new MySqlConnection("server=localhost;user id=root;password=1234;database=websitebansach");

return connect;

}

}

}

Lớp này để tạo kết nối với MySQL server.

1. Trong thư mục Models ta tạo một file Sach.cs chứ lớp Sach để mỗ tả dữ liệu của đối tượng sách.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebSiteBanSach.model

{

public class Sach

{

private int maSach;

private String tenSach;

private String tacGia;

private String hinhAnh;

private String moTa;

private int soLuongTon;

private int giaBan;

private int trangThai;

private int khuyenMai;

private int maTheLoai;

public int getMaTheLoai()

{

return maTheLoai;

}

public void setMaTheLoai(int maTheLoai)

{

this.maTheLoai = maTheLoai;

}

public Sach()

{

}

public int getMaSach()

{

return maSach;

}

public void setMaSach(int maSach)

{

this.maSach = maSach;

}

public String getTenSach()

{

return tenSach;

}

public void setTenSach(String tenSach)

{

this.tenSach = tenSach;

}

public String getTacGia()

{

return tacGia;

}

public void setTacGia(String tacGia)

{

this.tacGia = tacGia;

}

public String getHinhAnh()

{

return hinhAnh;

}

public void setHinhAnh(String hinhAnh)

{

this.hinhAnh = hinhAnh;

}

public String getMoTa()

{

return moTa;

}

public void setMoTa(String moTa)

{

this.moTa = moTa;

}

public int getSoLuongTon()

{

return soLuongTon;

}

public void setSoLuongTon(int soLuongTon)

{

this.soLuongTon = soLuongTon;

}

public int getGiaBan()

{

return giaBan;

}

public void setGiaBan(int giaBan)

{

this.giaBan = giaBan;

}

public int getTrangThai()

{

return trangThai;

}

public void setTrangThai(int trangThai)

{

this.trangThai = trangThai;

}

public int getKhuyenMai()

{

return khuyenMai;

}

public void setKhuyenMai(int khuyenMai)

{

this.khuyenMai = khuyenMai;

}

}

}

1. Tương tự ta tạo thư mục Dao trong project, trong thư mục đó tạo file SachDAO để truy xuất cơ sở dữ liệu với nội dung sau:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Collections;//Dùng cho reference ArrayList

using MySql.Data.MySqlClient;//Dùng cho Mysql

using WebSiteBansAch.Connect;//Dùng để gọi lớp DBConnect.cs(Khai báo namespace)

using WebSiteBansAch.Models;//Dùng để gọi lớp trong thư mục Models(Khai báo namespace)

//using System.Data;//Dùng cho reference DataSet

namespace WebsiteBansAch.Dao

{

public class SachDAO

{

public static List<Sach> getSachFromDB(String sql)

{

MySqlConnection connection = DBConnect.getConnection();

connection.Open();

MySqlCommand cmd = new MySqlCommand(sql, connection);

List<Sach> list = new List<Sach>();

MySqlDataReader rs = cmd.ExecuteReader();

while (rs.Read())

{

Sach sach = new Sach();

sach.setMaSach(rs.GetInt32("MaSach"));

sach.setTenSach(rs.GetString("TenSach"));

sach.setTacGia(rs.GetString("TacGia"));

sach.setHinhAnh(rs.GetString("HinhAnh"));

sach.setMoTa(rs.GetString("MoTa"));

sach.setSoLuongTon(rs.GetInt32("SoLuongTon"));

sach.setGiaBan(rs.GetInt32("GiaBan"));

sach.setTrangThai(rs.GetInt32("TrangThai"));

sach.setKhuyenMai(rs.GetInt32("KhuyenMai"));

sach.setMaTheLoai(rs.GetInt32("MaTheLoai"));

list.Add(sach);

}

rs.Close();

connection.Close();

return list;

}

public static List<Sach> getListSach()

{

String sql = "Select \* from Sach where TrangThai <> -1";

return getSachFromDB(sql);

}

}

}

1. Ta thay đổi nội dung ở HomeController như sau:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using WebsiteBansAch.Dao;

using WebSiteBansAch.Models;

namespace WebsiteBansAch.Controllers

{

public class HomeController : Controller

{

// GET: Home

public ActionResult Index()

{

List<Sach> list = SachDAO.getListSach();

ViewBag.listSach = list;

return View();

}

}

}

1. Ta thay đổi nội dung file Views/Home/Index.cshtml như sau:

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h3>Các đầu sách</h3>

<ul>

@foreach (var sach in ViewBag.listSach)

{

<div>

<img src="@Url.Content("~/img/"+sach.getHinhAnh())" alt="" style="height:120px; width:90px" />

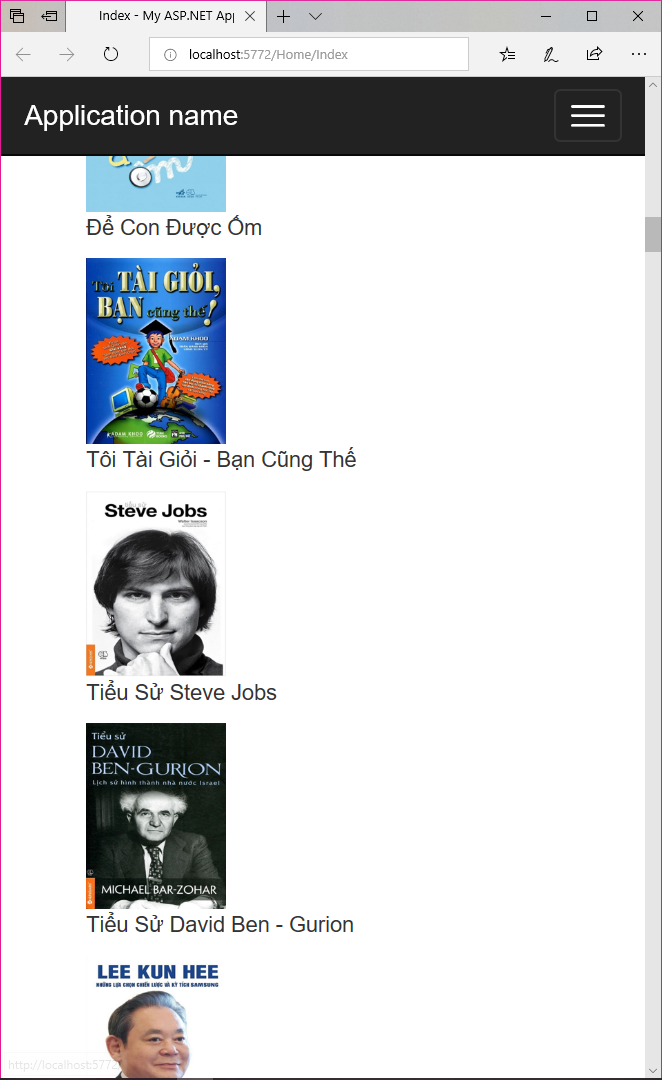
<p>@sach.getTenSach()</p>

</div>

}

</ul>

1. Tải hình ảnh các đầu sách [tại đây](https://drive.google.com/file/d/1r4BtoNC0I6fYHjqe1WKztRltE4YFxmlj/view?usp=sharing). Giải nén được thư mục chứa hình ảnh. Ta tạo thư mục tên img trong project sau đó chuột phải vào thư mục chọn **Add > Existing item > Chọn các file ảnh vừa giải nén và chọn OK**.
2. Nhấn F5 để chạy project ta được kết quả như hình.



# LAYOUT VÀ UI

## Nhúng CSS vào trang web:

### CSS là gì

**CSS (Cascading Style Sheets)** là một file có phần mở rông là **.css**, file này có tác dụng tách riêng phần định dạng (style) ra khỏi nội dung HTML.

### Sử dụng css:

1. Trong project ta tạo thư mục tên css
2. Trong thư mục css ta thêm file hcss-navbar.css với nội dung sau

@import url('https://fonts.googleapis.com/css?family=Roboto+Condensed00:900');

@import url('https://fonts.googleapis.com/css?family=Roboto:900,500');

.dropdown:hover .dropdown-menu {

display: block;

}

.dropdown-menu > li > a:focus, .dropdown-menu > li > a:hover {

background: none;

color: #FF5722;

}

.navbar-default .navbar-nav > .open > a, .navbar-default .navbar-nav > .open > a:focus, .navbar-default .navbar-nav > .open > a:hover {

color: #FF5722;

background-color: rgba(0,0,0,0.05);

}

.vl {

height: 30px;

width: 1px;

background: rgba(0,0,0,0.23);

}

.hcss-nav-logo {

height: 50px;

margin-top: 5px;

margin-left: 5px;

float: left;

}

.hcss-nav-button {

font-family: 'Roboto', sans-serif;

font-size: 15px;

font-weight: 500;

height: 100%;

}

.hcss-nav-button a {

color: #777;

}

.hcss-nav-button:hover {

color: #FF5722;

background-color: rgba(0,0,0,0.05);

transition: all 0.2s ease-in-out;

}

.hcss-nav-button a:hover {

color: #FF5722;

background-color: rgba(0,0,0,0.05);

transition: all 0.2s ease-in-out;

text-decoration: none;

}

.navbar-default .navbar-nav > li > a:hover {

color: #FF5722;

}

.hcss-nav-brand {

padding: 0;

margin: 0;

display: flex;

align-items: center;

margin-top: 0px;

}

.hcss-nav-brandname {

font-family: 'Roboto', sans-serif;

color: #FF5722;

display: inline-block;

font-weight: 900;

font-size: 23px;

margin-left: 5px;

}

.search-text {

background-color: transparent;

font-weight: bold;

height: 34px;

color: rgba(0,0,0,0.59);

border: none;

}

.search-text:focus {

background: none;

box-shadow: none;

}

.search-button {

background-color: transparent;

font-weight: bold;

height: 34px;

color: rgba(0,0,0,0.19);

border: none;

}

.search-button:hover {

color: rgba(252,94,53,1);

background: none;

box-shadow: none;

}

.cart {

float: right;

width: 25px;

margin-right: 20px;

}

.search-form {

margin-right: 30px;

width: 50%;

float: right;

}

.search-form-border {

border: 1.5px solid #ccc;

border-radius: 5px;

width: 240px;

}

.search-form-border:hover {

border: 1.5px solid rgba(252,94,53,1);

}

.space {

margin-left: 10px;

}

.navbar-header {

display: flex;

align-items: center;

}

@media screen and (max-width: 320px) {

.search-form {

padding: 10px;

width: 100%;

margin: 0;

}

.navbar-xs {

display: flex;

margin-right: 2px;

}

.search-form-border {

width: 160px;

margin-right: 5px;

}

.cart {

margin-right: 10px;

}

.space {

margin-left: 0px;

}

}

@media screen and (min-width: 321px) and (max-width: 375px) {

.search-form {

width: 45%;

}

.navbar-xs {

display: flex;

margin-right: 2px;

}

.search-form-border {

width: 200px;

margin-right: 5px;

}

.space {

margin-left: 0px;

}

.cart {

margin-right: 10px;

}

}

/\* #### Mobile Phones Portrait or Landscape #### \*/

@media screen and (min-width: 376px) and (max-width: 640px) {

.hcss-nav-brand {

letter-spacing: 2px;

font-weight: 600;

}

.cart {

margin-left: 12px;

}

.navbar-xs {

display: flex;

margin-right: 2px;

}

.space {

margin-left: 0px;

}

.search-form-border {

width: 230px;

}

}

@media screen and (min-width: 641px) and (max-width: 767px) {

.cart {

float: right;

width: 23px;

margin-top: 15px;

margin-bottom: 10px;

margin-right: 30px;

}

.navbar-nav {

margin: 0;

text-align: center;

}

}

/\* #### Tablets Portrait or Landscape #### \*/

@media screen and (min-width: 768px) and (max-width: 1023px) {

.navbar-header {

display: flex;

align-items: center;

width: 100%;

}

.navbar-nav {

float: left;

margin: 0;

width: 100%;

justify-content: space-evenly;

display: flex;

}

.navbar .navbar-collapse {

text-align: center;

}

.cart {

float: right;

width: 20px;

margin-right: 26px;

}

.navbar-nav > li > .dropdown-menu {

margin-top: 10px;

background-color: white;

border-radius: 5px;

float: left;

width: 600px;

margin-left: -200px;

padding: 10px;

}

.dropdown-item {

height: 35px;

align-items: center;

display: flex;

}

.dropdown-item > a {

text-decoration: none;

font-family: 'Roboto', sans-serif;

font-size: 15px;

font-weight: 500;

position: absolute;

left: 10px;

transition: all 0.2s ease-in-out;

}

.dropdown-item:hover {

background-color: rgba(0,0,0,0.05);

justify-content: center;

transition: all 0.2s ease-in-out;

border-radius: 5px;

}

.dropdown-item:hover > a {

color: #FF5722;

position: absolute;

left: 30%;

transform: translateX(-20%);

}

}

/\* #### Desktops #### \*/

@media screen and (min-width: 1024px) {

.navbar .navbar-nav {

margin-left: 10px;

display: flex;

align-items: center;

}

.navbar .navbar-collapse {

text-align: center;

}

.navbar-header {

padding: 0;

}

.right {

float: right;

}

.left {

float: left;

}

.dropdown:hover .dropdown-menu {

display: block;

}

.navbar-nav > li > .dropdown-menu {

border-color: transparent;

background-color: white;

border-radius: 5px;

float: left;

width: 600px;

margin-left: -200px;

padding: 10px;

}

.dropdown-item {

height: 35px;

align-items: center;

display: flex;

}

.dropdown-item > a {

text-decoration: none;

font-family: 'Roboto', sans-serif;

font-size: 15px;

font-weight: 500;

position: absolute;

left: 10px;

transition: all 0.2s ease-in-out;

}

.dropdown-item:hover {

background-color: rgba(0,0,0,0.05);

justify-content: center;

transition: all 0.2s ease-in-out;

border-radius: 5px;

}

.dropdown-item:hover > a {

color: #FF5722;

position: absolute;

left: 30%;

transform: translateX(-20%);

}

}

1. Tại file **Views/Home/Header.cshtml** trong thẻ head ta thêm đoạn mã sau để nhúng file css vừa tạo ở trên.

<link href="~/css/hcss-navbar.css" rel="stylesheet" type="text/css" media="all" />

1. Trong thẻ body ở Header.cshtml ta có thể sử dụng class css trong file vừa linkví dụ như thẻ a dưới đây.

<a href="~/Home/index" class="hcss-nav-brand navbar-brand">

<img src="~/images/LOGO.png" class="hcss-nav-logo" />

<div class="hcss-nav-brandname hidden-xs">ABC</div>

</a>

## Reponsive design với boostrap

### Boostrap la gì?

Bootstrap là một framework HTML, CSS, và JavaScript cho phép thiết kế phát triên responsive web mobile. Nếu bạn chưa hiểu về responsive web mobile bạn có thể xem bài viết của tôi ở đây để hiểu hơn về nó “Khác nhau cơ bản giữa mobile web và Responsive Web Design“.

### Sử dụng boostrap 3

Thêm Bootstrap từ CDN

<link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.css"> <!-- Optional theme --> <link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap-theme.min.css"> <!-- Latest compiled and minified JavaScript --> <script src="//netdna.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js"></script>

Chèn các class boostrap vào project:

<div class="col-lg-2 col-md-3 col-sm-3 col-xs-6">

<div class="col-md hcss-sachkm-sach hcss-custom-slideshow">

<div class="col-md hcss-sachkm-sach hcss-custom-slideshow">

...

</div>

</div>

</div>

# SESSION

## Cách sử dụng session:

Để khách hàng có thể vào được một số trang xem thông tin liên quan đến cá nhân hoặc tiến hành thanh toán đơn hàng thì họ phải đăng nhập vào hệ thống.

Khi khách hàng nhập username và password để đăng nhập vào hệ thống. Hệ thống sẽ kiểm tra trong database nếu tồn tại khách hàng mà có có username và password trùng với các thông tin nhập vào, sau đó sẽ tạo một Session có SessionID là user. Session lưu lại đối tượng khách hàng đã tìm thấy. Lúc này bạn có thể vào các trang để xem các thông tin có liên quan đến user này.

Cách tạo Session trong Asp.Net rất đơn giản.

Như ví dụ trên, ta có thể tạo session để lưu lại đối tượng khách hàng khi đăng nhập:

kh = KhachHangDAO.loginKhachHang(f["email"].ToString(), MD5.encryption(f["pass"].ToString()));

Session["user"] = kh;

Đồng thời khi khách hàng đăng ký tài khoản trên hệ thống, hệ thống thêm khách hàng mới vào database, sau đó tạo một Session để lưu lại đối tượng khách hàng vừa đăng ký.

Cách tạo Session để lưu lại đối tượng khách hàng khi đăng ký:

kh = new KhachHang();

kh.setHoTenKhachHang(f["username"]);

kh.seteMail(f["email"]);

kh.setMatKhau(MD5.encryption(f["pass"]));

KhachHangDAO.insertKhachHang(kh);

Session["user"] = kh;

Session được dùng để lưu trữ thông tin liên quan đến người dùng, nhưng thay vì được lưu trên máy client, các biến session được lưu trên máy server. Mỗi session có một định danh duy nhất (SessionID).

Thông thường, đối tượng session sẽ tự hủy khi đóng ứng dụng hoặc khi hết thời gian timeout. Thời gian timeout này mặc định là 20 phút. Bạn có thể set lại thời gian timeout của session trong file Web.config của Asp.Net.

<configuration>

<system.web>

<sessionState timeout="20"></sessionState>

</system.web>

</configuration>

# FACEBOOK COMMENT PLUGIN

## Tại sao nên sử dụng Facebook comment plugin:

Việc nhúng Facebook Comment vào website là công cụ tuyệt vời giúp bạn tương tác với khách hàng - đa số khách hàng đều có tài khoản trên mạng xã hội Facebook, qua đó bạn có thêm nhiều traffic, giữ chân khác ở lại website lâu hơn.

Bài viết có nhiều comment chứng tỏ được nhiều người quan tâm, sự tin tưởng cũng sẽ tăng lên.

Người dùng không phải đăng nhập vào website để bình luận vì đã có sẵn tài khoản Facebook.

Không tốn công code, lưu trữ và quản lý các bình luận.

## Cách nhúng Facebook comment plugin vào website

1. Nếu chưa có tài khoản Facebook thì bạn hãy đăng ký tài khoản, sau đó tạo facebook developer để có thể tích hợp bình luận vào website.
2. Truy cập vào <https://developers.facebook.com/apps> rồi chọn **Add new APP**

A screenshot of a social media post

Description generated with very high confidence

1. A screenshot of a cell phone

   Description generated with very high confidenceAdd product Facebook Login
2. Tiếp theo bạn search Facebook Comment trên thanh tìm kiếm hiển thị kết quả như hình bên dưới. Sau đó chọn bạn sẽ chọn Comments-Social Plugins.
3. A screenshot of a social media post

   Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidence

1. Dán đoạn mã SDK JavaScript vào phần thẻ <head>

<!--Nhúng facebook comment -->

<div id="fb-root"></div>

<script>

(function (d, s, id) {

var js, fjs = d.getElementsByTagName(s)[0];

if (d.getElementById(id))

return;

js = d.createElement(s);

js.id = id;

js.src = "//connect.facebook.net/en\_GB/sdk.js#xfbml=1&version=v2.8&appId=233524923783770";

fjs.parentNode.insertBefore(js, fjs);

}(document, 'script', 'facebook-jssdk'));

</script>

1. Tiếp theo, ở chỗ nào muốn hiển thị phần bình luận bạn sẽ dán đoạn mã thứ 2 của Facebook cung cấp. Ở mục href là đường dẫn đến trang muốn đặt ô comment.

<!--Nhúng vào facebook-->

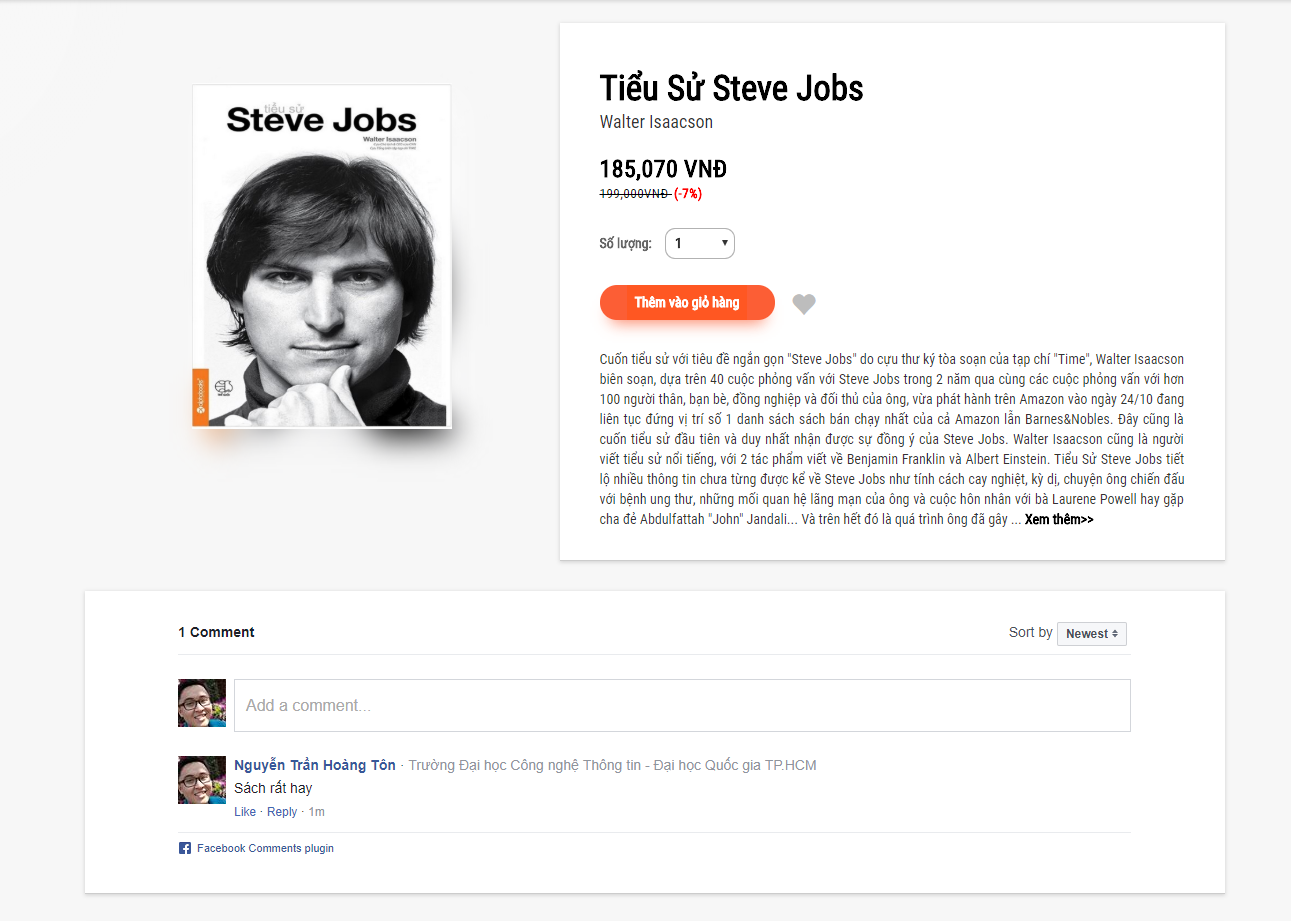
<div class="hcss-facebookaddon">

<div class="fb-comments" data-href="http://localhost:4527/Sach/single?maSach=@sach.getMaSach()" data-width="100%" data-numposts="5">

</div>

</div>

Như vậy là đã xong phần tích hợp Facebook vào website , bạn hoàn toàn có thể quản lý được phần bình luận mà người dùng đăng.



# Các thành phần khác

## Ajax

### Ajax là gì:

AJAX - "Asynchronous JavaScript and XML" - là một bộ công cụ cho phép load dữ liệu từ server mà không yêu cầu tải lại trang. Nó sử dụng chức năng sẵn có XMLHttpRequest(XHR) của trình duyệt để thực hiện một yêu cầu đến server và xử lý dữ liệu server trả về.

### Sử dụng Ajax:

Ajax không phải là ngôn ngữ mà là kĩ thuật trong Javascript, giúp “load dữ liệu không cần load lại trang” hay “load một phần website”, giúp website linh hoạt, nhanh chóng và thân thiện với người dùng hơn.

Để sử dụng kĩ thuật Ajax trong jQuery, trước hết các bạn cần thư viện jQuery được tải về từ <http://jquery.com/download>

Đoạn code phía dưới ví dụ sử dụng Ajax khi người dùng like một cuốn sách.

<script>

$(document).on('click', '#btn-like', function (e) {

$.ajax({

url: '/BRKhachHang/Save',

dataType: "json",

data: { maKhachHang: jmaKhachHang, turn: turn, maSach: jmaSach },

type: "POST",

success: function (result) {

},

error: function (responseText) {

alert("error");

}

});

});

</script>

Trong cặp thẻ “<script>” ta có:

$(document).on('click', '#btn-like', function (e){…}): có nghĩa là code của bạn chỉ được thực thi khi bạn click vào button yêu thích có id là btn-like tại mỗi cuốn sách.

Đối số đầu tiên chúng ta truyền vào cho phương thức “ ajax() ” chính là một đối tượng (Object) gồm các thuộc tính cấu hình để kĩ thuật AJAX của chúng ta có thể thực thi. Trong đó:

**url:** chuỗi chứa đường dẫn tới controller có action tương ứng để xử lý dữ liệu. Cách gọi tới hàm xử lý: Controller/Action

**type:** phương thức gửi đi tương tự như của “ <form> ”, mặc định là “GET” nếu như các bạn không truyền vào. Ở đoạn code trên ta sử dụng phương thức “POST”

**dataType:** xác định dữ liệu trả về thuộc dạng nào? Ở đoạn code trên truyền vào kiểu dữ liệu json.

**data:** Dữ liệu bạn sẽ truyền sang đường dẫn chỉ định để thực hiện xử lý và trả về dữ liệu.

**“ success ” và “error”:** Tương ứng việc thực thi thành công hay thất bại.

## jQuery:

### jQuere là gì:

jQuery là 1 bộ thư viện được viết dựa trên ngôn ngữ JavaScript giúp đơn giản hoá việc viết mã lệnh JavaScript.

jQuery là thư viện mã nguồn mở (hoàn toàn miễn phí sử dụng) được bắt đầu phát triển **John Resig** và phiên bản đầu tiên được ra đời vào tháng 8 năm 2006.

### Sử dụng jQuery

Trong project này, jquery được sử dụng cùng với ajax (mục 6.1) và được dùng để tính số lượng, tổng tiền trong giỏ hàng

1. Đầu tiên ta thêm thẻ sau vào thẻ head

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

1. Ta thêm đoạn code sau để cập nhật thông tin giỏ hàng

<script>

$('.addnumber').click(function () {

var tr = $(this).closest('tr');

console.log(tr);

var soluong = getNumber(tr, '#soluong');

var dongia = getNumber(tr, '#dongia');

var khuyenmai = 1;

var thanhtien = getNumber(tr, '#thanhtien');

var tongtien = parseInt($('#tongtien').text().replace(/[$,]+/g, ""));

var tongsoluong = parseInt($('#tongsoluong').text().replace(/[$,]+/g, ""));

var maSach = parseInt(tr.find('#maSach').text());

console.log(soluong, dongia, khuyenmai, thanhtien, tongtien);

if (soluong < getNumber(tr, '#soluongton')) {

soluong = soluong + 1;

thanhtien = thanhtien + dongia \* khuyenm ai;

tongtien = tongtien + dongia \* khuyenmai;

tongsoluong += 1;

}

console.log(soluong, dongia, khuyenmai, thanhtien, tongtien);

//Save Ajax

$.ajax({

url: '/Cart/capnhatgiohang',

dataType: "json",

data: { maSach: maSach, soLuongMua: soluong },

type: "GET",

success: function (result) {

},

error: function (responseText) {

alert("error");

}

});

tr.find('#soluong').html(soluong);

tr.find('#thanhtien').html(thanhtien.format(0, 3, ',', '') + " VNĐ");

$('#tongtien').html(tongtien.format(0, 3, ',', '') + " VNĐ");

$('#tongsoluong').html(tongsoluong);

if (tr.find('.cbutton--effect-boris').hasClass('red'))

tr.find('.cbutton--effect-boris').removeClass('red');

tr.find('.cbutton--effect-boris').toggleClass('green');

setTimeout(function () {

$this.removeClass('green')

}, 300);

var $this = $(this);

if ($this.hasClass('cbutton--click')) {

$this.removeClass('cbutton--click')

} else {

$this.addClass('cbutton--click')

setTimeout(function () {

$this.removeClass('cbutton--click')

}, 300);

}

});

$('.subnumber').click(function () {

var tr = $(this).closest('tr');

console.log(tr);

var soluong = getNumber(tr, '#soluong');

var dongia = getNumber(tr, '#dongia');

var khuyenmai = 1;

var thanhtien = getNumber(tr, '#thanhtien');

var tongtien = parseInt($('#tongtien').text().replace(/[$,]+/g, ""));

var tongsoluong = parseInt($('#tongsoluong').text().replace(/[$,]+/g, ""));

var maSach = parseInt(tr.find('#maSach').text());

console.log(soluong, dongia, khuyenmai, thanhtien, tongtien);

if (soluong > 1) {

soluong = soluong - 1;

thanhtien = thanhtien - dongia \* khuyenmai;

tongtien = tongtien - dongia \* khuyenmai;

tongsoluong -= 1;

}

console.log(soluong, dongia, khuyenmai, thanhtien, tongtien);

//Save Ajax

$.ajax({

url: '/Cart/capnhatgiohang',

dataType: "json",

data: { maSach: maSach, soLuongMua: soluong },

type: "GET",

success: function (result) {

},

error: function (responseText) {

alert("error");

}

});

tr.find('#soluong').html(soluong);

tr.find('#thanhtien').html(thanhtien.format(0, 3, ',', '') + " VNĐ");

$('#tongtien').html(tongtien.format(0, 3, ',', '') + " VNĐ");

$('#tongsoluong').html(tongsoluong);

if (tr.find('.cbutton--effect-boris').hasClass('green'))

tr.find('.cbutton--effect-boris').removeClass('green');

tr.find('.cbutton--effect-boris').toggleClass('red');

setTimeout(function () {

$this.removeClass('red')

}, 300);

var $this = $(this);

if ($this.hasClass('cbutton--click')) {

$this.removeClass('cbutton--click')

} else {

$this.addClass('cbutton--click')

setTimeout(function () {

$this.removeClass('cbutton--click')

}, 200);

}

});

</script>

## Google Maps:

### Google maps là gì

Google Map là một dịch vụ ứng dụng vào công nghệ bản đồ trực tuyến trên web miễn phí được cung cấp bởi Google, hỗ trợ nhiều dịch vụ khác của Google đặc biệt là dò đường và chỉ đường; hiển thị bản đồ đường sá, các tuyến đường tối ưu cho từng loại phương tiện, cách bắt xe và chuyển tuyến cho các loại phương tiện công cộng (xe bus, xe khách ...), và những địa điểm (kinh doanh, trường học, bệnh viện, cây ATM...) trong khu vực cũng như khắp nơi trên thế giới.

### Map Api

Là một phương thức cho phép 1 website B sử dụng dịch vụ bản đồ của website A (gọi là Map API) và nhúng vào website của mình (site B). Site A ở đây là google map, site B là các website cá nhân hoặc tổ chức muốn sử dụng dịch vụ của google (di chuột, room, đánh dấu trên bản đồ…)

Các ứng dụng xây dựng trên maps được nhúng vào trang web cá nhân thông qua các thẻ javascripts do vậy việc sử dụng API google rất dễ dàng.

### Sử dụng

Trong project này, map api được sử dụng để hiện thị địa chỉ trụ sở Website

Đoạn code phía dưới thể hiện việc nhúng map api vào footer của trang web



# KẾT LUẬN

* ASP.NET MVC là một framework phát triển web mạnh mẽ giúp lập trình viên có thể hoàn toàn kiểm soát HTML, CSS và JavaScript để tùy biến mọi thứ. Không bị bó hẹp như webform.
* Dùng mô hình MVC để chia ứng dụng ASP.NET MVC ra làm 3 phần chính là Model, View, và Controller điều này làm cho việc phát triển các ứng dụng lớn có độ phức tạp cao được dễ dàng hơn và dễ bào trì hơn.
* Có thể tích hợp những thư viện hiện đại như Angular, Bootstrap hoặc jQuery.
* Ngoài ra, hệ thống định tuyến (routing system) của ASP.NET MVC giúp nó tùy chỉnh đường dẫn URL cho thân thiện với người dùng. Đặc biệt, ta được phép cấu hình URL theo bất kì khuôn mẫu nào ta thấy vừa ý.