**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT CÔNG NGHỆ**

****

**TIỂU LUẬN THỰC TẬP DOANH NGHIỆP**

**TÌM HIỂU VỀ LẬP TRÌNH WEBSITE TRÊN NỀN TẢNG ASP.NET MVC**

**Ngành : Công nghệ thông tin**

**Chuyên ngành : Kỹ thuật phần mềm**

**Giảng viên hướng dẫn : Ths. Trần Văn Hữu**

**Nhóm thực hiện : Nhóm 6**

**Sinh viên: Tăng Trung Hiếu - 1724801030044**

**Lê Văn Đông - 1724801030028**

***Bình Dương , Tháng 11/2020***

Mục Lục

[Mục Lục ii](#_Toc58231697)

[DANH SÁCH CÁC KÝ TỰ, CÁC CHỮ VIẾT TẮT 1](#_Toc58231698)

[Chương I: GIỚI THIỆU VÀ CÀI ĐẶT 2](#_Toc58231699)

[1. Giới thiệu công nghệ 2](#_Toc58231700)

[1.1. ASP.NET là gì 2](#_Toc58231701)

[1.2. ASP.NET MVC là gì 2](#_Toc58231702)

[1.3. Kiến trúc MVC 3](#_Toc58231703)

[1.4. Một số tính năng nổi bật của ASP.Net MVC 3](#_Toc58231704)

[2. Hướng dẫn cài đặt 4](#_Toc58231705)

[3. Cấu trúc thư mục MVC ASP.NET 11](#_Toc58231706)

[3.1.App\_Data 11](#_Toc58231707)

[3.2. App\_Start 11](#_Toc58231708)

[3.3. Content 12](#_Toc58231709)

[3.4. Controller 13](#_Toc58231710)

[3.5. Fonts 14](#_Toc58231711)

[3.6. Models 14](#_Toc58231712)

[3.7. Scripts 14](#_Toc58231713)

[3.8. Views 16](#_Toc58231714)

[4. Controller 17](#_Toc58231715)

[4.1. Trách nhiệm của Controller 17](#_Toc58231716)

[4.2. Thêm mới Controller 17](#_Toc58231717)

[5. Models 21](#_Toc58231718)

[Cách thêm một Model trong ASP.NET MVC 21](#_Toc58231719)

[6. Views 22](#_Toc58231720)

[7. Phương thức Action 27](#_Toc58231721)

[8. Tích hợp Controller , Models và Views 28](#_Toc58231722)

[9. Model Binding 33](#_Toc58231723)

[Chương II: PHÂN TÍCH DỰ ÁN 35](#_Toc58231724)

[1. Các biên bản họp nhóm 35](#_Toc58231725)

[2. Danh sách các chức năng đề tài cần đáp ứng 37](#_Toc58231726)

[3. Danh sách chức năng đề tài không làm 37](#_Toc58231727)

[4. Công cụ để làm đề tài 38](#_Toc58231728)

[5. Cấu hình tối thiểu để chạy được sản phẩm của đề tài 38](#_Toc58231729)

[6. Các biểu đồ Use Case của đề tài 38](#_Toc58231730)

[Chương III: Thiết kế cơ sở dữ liệu 41](#_Toc58231731)

[3.1 Bảng dữ liệu 41](#_Toc58231732)

[3.1.1 Bảng dữ liệu MenuType 41](#_Toc58231733)

[3.1.2 Bảng dữ liệu menu 43](#_Toc58231734)

[3.1.3 Bảng dữ liệu GameCategory 44](#_Toc58231735)

[3.1.4 Bảng dữ liệu Game 45](#_Toc58231736)

[3.1.5 Bảng dữ liệu TaiKhoan 46](#_Toc58231737)

[3.1.6 Bảng dữ liệu KhachHang 47](#_Toc58231738)

[3.2 Diagram 48](#_Toc58231739)

[3.3 Biểu đồ lớp 48](#_Toc58231740)

[Chương IV: TRIỂN KHAI , ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN 49](#_Toc58231741)

[1. Các chức năng đã làm được 49](#_Toc58231742)

[1.1. Chức năng đăng nhập 49](#_Toc58231743)

[2. Chức năng chính 50](#_Toc58231744)

[2.1. Giao diện dành cho người quản trị 50](#_Toc58231745)

[2.2. Giao diện Quản Lý Game 51](#_Toc58231746)

[2.3. Giao diện Quản Lý Menu 52](#_Toc58231747)

[2.4. Giao diện Dành cho khách hàng 53](#_Toc58231748)

[2.5. Giao diện Game theo thể loại 54](#_Toc58231749)

[2.6. Giao diện Chi tiết Game 54](#_Toc58231750)

[2.7. Giao diện tìm kiếm Game 55](#_Toc58231751)

[2. Những khó khăn và hạn chế trong quá trình thực hiện đề tài 56](#_Toc58231752)

[3. Hướng phát triển của đề tài 56](#_Toc58231753)

[TÀI LIỆU THAM KHẢO 56](#_Toc58231754)

# DANH SÁCH CÁC KÝ TỰ, CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| Từ viết tắt | Giải thích |
| PK | Khóa chính (Primary Key ) |

# Chương I: GIỚI THIỆU VÀ CÀI ĐẶT

### 1. Giới thiệu công nghệ

### 1.1. ASP.NET là gì

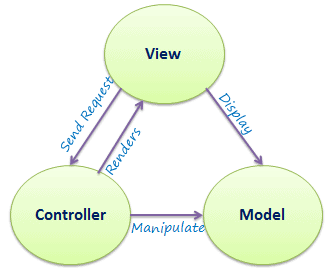
* ASP.NET là một nền tảng ứng dụng web (web application framework) được phát triển và cung cấp bởi Microsoft, cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web. ASP.NET đã tích hợp HTML, CSS, Javascript . Lần đầu tiên được đưa ra thị trường vào tháng 2 năm 2002 cùng với phiên bản 1.0 của.NET framework, là công nghệ nối tiếp của Microsoft's Active Server Pages(ASP). 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

### 1.2. ASP.NET MVC là gì

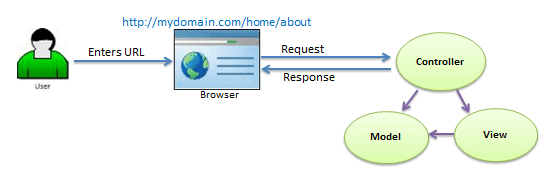
* ASP.NET MVC là một framework tuyệt vời hỗ trợ pattern MVC cho ASP.NET. Nếu bạn muốn hiểu ASP.NET MVC làm việc như thế nào, bạn cần phải có một sự hiểu biết rõ ràng về mô hình MVC. MVC là cụm từ viết tắt của Model-View-Controller, nó phân chia pattern của ứng dụng thành 3 phần - model, controller và view.
* Model giúp lưu trữ dữ liệu của ứng dụng và trạng thái của nó. Nó là một cơ sở dữ liệu hoặc cũng có thể chỉ là một tập tin XML.
* View được coi là một giao diện người dùng được sử dụng bởi khách truy cập trang web của bạn để nhìn thấy các dữ liệu. Các trang ASPX thường được sử dụng để hiển thị view trong các ứng dụng ASP.NET MVC.
* Controller chịu trách nhiệm xử lý các tương tác của người dùng với trang web. Nó được sử dụng để xác định loại view nào cần phải được hiển thị. Controller cũng được sử dụng cho mục đích giao tiếp với model

### 1.3. Kiến trúc MVC

Hình dưới đây minh họa sự tương tác giữa Model, View và Controller.



Hình dưới đây minh họa luồng yêu cầu của người dùng trong ASP.NET MVC.



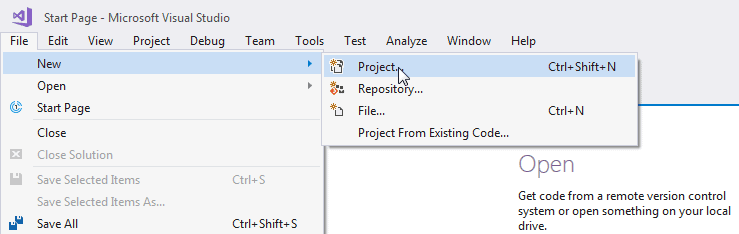
### 1.4. Một số tính năng nổi bật của ASP.Net MVC

* Tách bạch các tác vụ của ứng dụng (logic nhập liệu, business logic, và logic giao diện), dễ dàng kiểm thử và mặc định áp dụng hướng phát triển TDD. Tất cả các tính năng chính của mô hình MVC được cài đặt dựa trên interface và được kiểm thử bằng cách sử dụng các đối tượng mocks, mock object là các đối tượng mô phỏng các tính năng của những đối tượng thực sự trong ứng dụng. Bạn có thể kiểm thử unit-test cho ứng dụng mà không cần chạy controller trong tiến trình ASP.NET, và điều đó giúp unit test được áp dụng nhanh chóng và tiện dụng. Bạn có thể sử dụng bất kỳ nền tảng unit-testing nào tương thích với nền tảng .NET.
* MVC là một nền tảng khả mở rộng (extensible) & khả nhúng (pluggable). Các thành phần của ASP.NET MVC được thiết kể để chúng có thể được thay thế một cách dễ dàng hoặc dễ dàng tùy chỉnh. Bạn có thể nhúng thêm view engine, cơ chế định tuyến cho URL, cách kết xuất tham số của action-method và các thành phần khác. ASP.NET MVC cũng hỗ trợ việc sử dụng Dependency Injection (DI) và Inversion of Control (IoC). DI cho phép bạn gắn các đối tượng vào một lớp cho lớp đó sử dụng thay vì buộc lớp đó phải tự mình khởi tạo các đối tượng. IoC quy định rằng, nếu một đối tượng yêu cầu một đối tượng khác, đối tượng đầu sẽ lấy đối tượng thứ hai từ một nguồn bên ngoài, ví dụ như từ tập tin cấu hình. Và nhờ vậy, việc sử dụng DI và IoC sẽ giúp kiểm thử dễ dàng hơn.
* ASP.NET MVC có thành phần ánh xạ URL mạnh mẽ cho phép bạn xây dựng những ứng dụng có các địa chỉ URL xúc tích và dễ tìm kiếm. Các địa chỉ URL không cần phải có phần mở rộng của tên tập tin và được thiết kế để hỗ trợ các mẫu định dạng tên phù hợp với việc tối ưu hóa tìm kiếm (URL) và phù hợp với lập địa chỉ theo kiểu REST..
* Hỗ trợ các tính năng có sẵn của ASP.NET như cơ chế xác thực người dùng, quản lý thành viên, quyền, output caching và data caching, seession và profile, quản lý tình trạng ứng dụng, hệ thống cấu hình…
* ASP.NET MVC sử dụng view engine Razor View Engine cho phép thiết lập các view nhanh chóng, dễ dàng và tốn ít công sức hơn so với việc sử dụng Web Forms view engine.

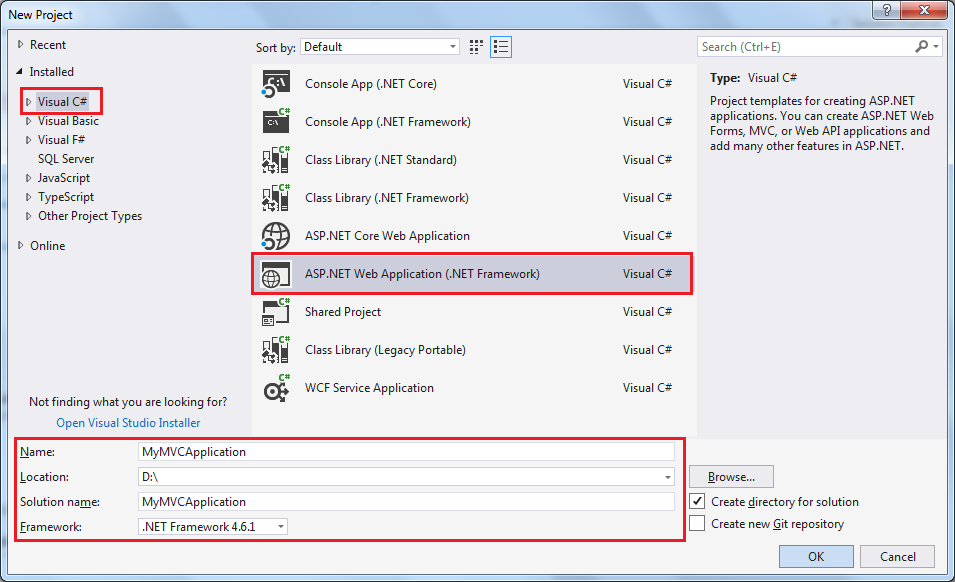
## 2. Hướng dẫn cài đặt

* **Bước 1:**Tại đây, chúng ta sẽ sử dụng MVC v5.2, phiên bản Cộng đồng Visual Studio 2017 và .NET framework 4.6 để tạo ứng dụng MVC đầu tiên của chúng tôi.

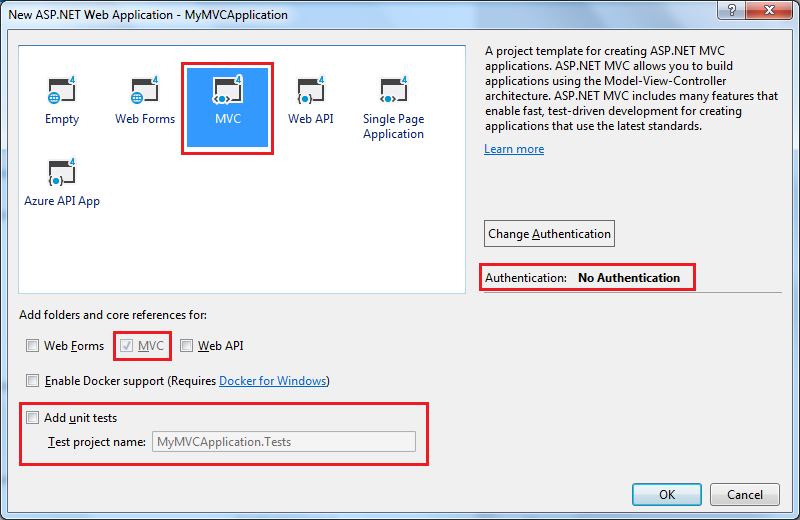
Mở  Visual Studio 2017  và chon  **File menu** -> **New** -> **Project** :



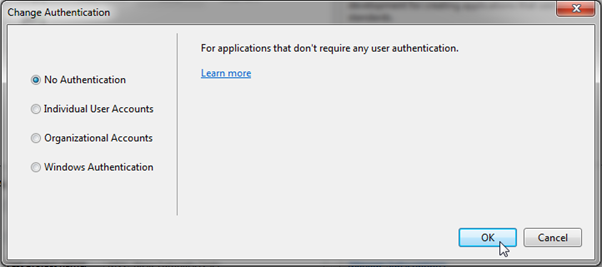
* **Bước 2**: Từ hộp thoại **New Project,** mở rộng Visual C#  và chọn **Web,**sau đó chọn **ASP.NET Web Application (.NET Framework)** . Nhập tên project  MyMVCApplication. Chúng ta cũng có thể thay đổi vị trí lưu project bằng cách vào  **Browse..**. Cuối cùng, click **OK.**



* **Bước 3**: Từ hộp thoại New ASP.NET Web Application, chọn MVC:

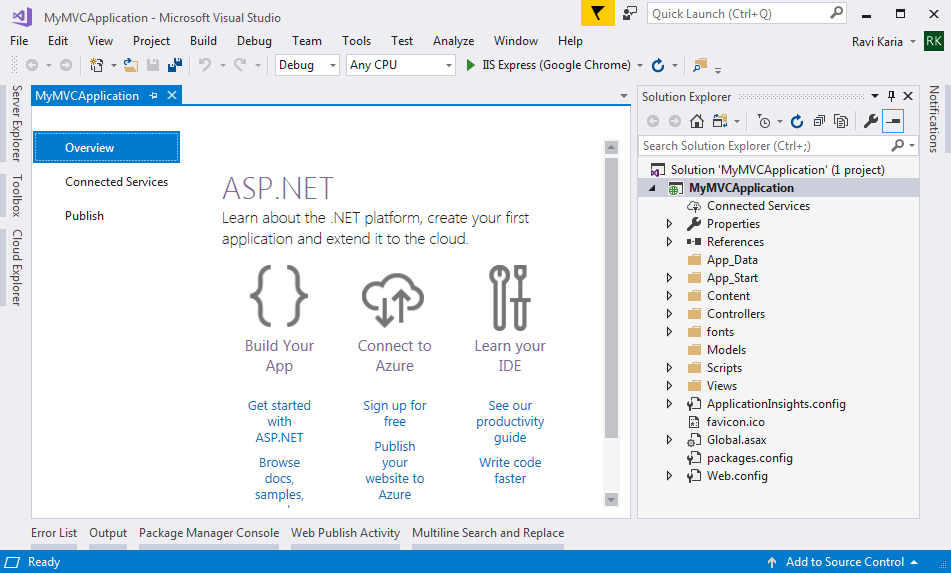


Chúng ta cũng có thể thay đổi quyền bằng cách click **Change Authentication**button.

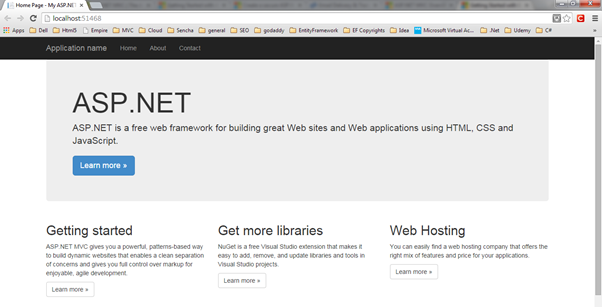


* **Bước 4 (Chọn Loại xác thực)**: Ở đây, chúng ta đang giữ xác thực mặc định cho ứng dụng. Ứng dụng của tôi là Không xác thực. Nhấn **OK**để tiếp tục.

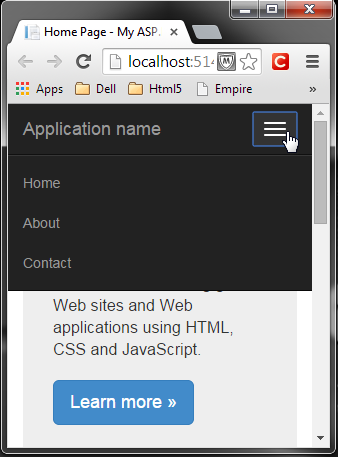
Đợi một lúc cho đến khi Visual Studio tạo một project  MVC đơn giản bằng cách sử dụng mẫu mặc định như dưới đây.



Bây giờ , nhấn F5 to run một project in debug mode hoặc Ctrl + F5 to run một project không cần deug. Khi chạy nó sẽ mở trang chủ trong trình duyệt.



Proejct  MVC 5 đã chứa các tệp JavaScript và CSS bootstrap 3.0 mặc định. Vì vậy, bạn có thể tạo các trang web nhanh. Giao diện người dùng  sẽ thay đổi giao diện dựa trên kích thước màn hình của các thiết bị khác nhau. Ví dụ, thanh menu trên cùng sẽ được thay đổi trong các thiết bị di động.



Kết quả trả về trong ứng dụng MVC :

**HomeController.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace MvcApplicationDemo.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

return View();

}

public ActionResult About()

{

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

ViewBag.Message = "Your contact page.";

return View();

}

}

}

**Index.cshtml**

@{

ViewBag.Title = "Home Page";

}

<div class="jumbotron">

<h1>ASP.NET</h1>

<p class="lead">ASP.NET is a free web framework for building great Web sites and Web applications

using HTML, CSS and JavaScript.</p>

<p><a href="https://asp.net" class="btn btn-primary btn-lg">Learn more</a></p>

</div>

<div class="row">

<div class="col-md-4">

<h2>Getting started</h2>

<p>

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that

enables a clean separation of concerns and gives you full control over markup

for enjoyable, agile development.

</p>

<p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301865">

Learn more</a></p>

</div>

<div class="col-md-4">

<h2>Get more libraries</h2>

<p>NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries

and tools in Visual Studio projects.</p>

<p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301866">

Learn more</a></p>

</div>

<div class="col-md-4">

<h2>Web Hosting</h2>

<p>You can easily find a web hosting company that offers the right mix of features and price

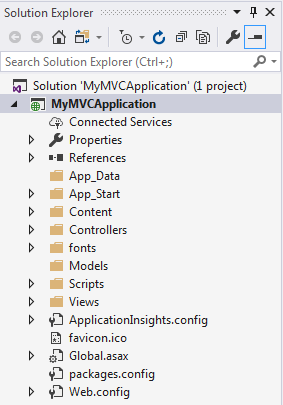
for your applications.</p>

<p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301867">

Learn more</a></p>

</div> </div>

## 3. Cấu trúc thư mục MVC ASP.NET

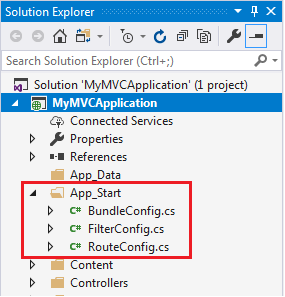


### ****3.1.App\_Data****

App\_Data: thư mục này chứa dữ liệu như: LocalDB, .mdf files, xml files. IIS sẽ không bao giờ nhận các tệp từ thư mục App\_Data.

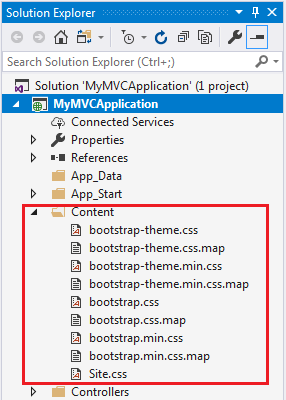
### ****3.2.**** ****App\_Start****

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, tương tự như Middleware của PHP. Và file **RouteConfig.cs**, cái tên đã nói lên tất cả, chúng ta sẽ định nghĩa các routes của web ở trong file này.



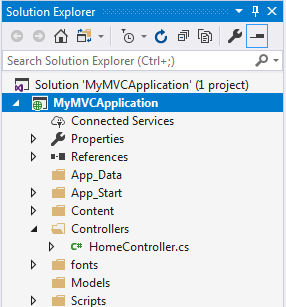
### ****3.3.**** ****Content****

Content: thư mục này chứa các file :css, images và icons .Ứng dụng  MVC 5 chứa các file mặc định: bootstrap.css, bootstrap.min.css và Site.css .

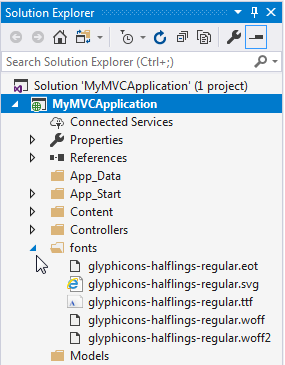


### ****3.4.**** ****Controller****

Controllers:Tất nhiên là nó dùng để chứa controller rồi. Controller xử lý yêu cầu và phản hồi cho người dùng. MVC yêu cầu tên của tất cả các tệp điều khiển kết thúc bằng "Controller". Bạn sẽ tìm hiểu về Controller trong phần tiếp theo.



### ****3.5.**** ****Fonts****

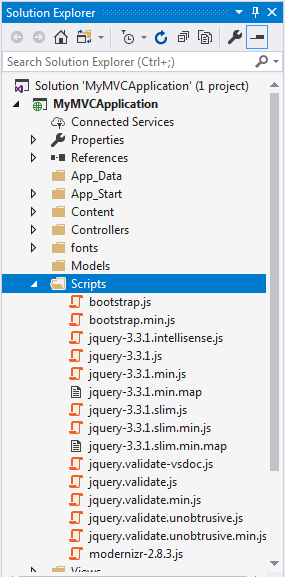


### ****3.6.**** ****Models****

Models chứa các file tương tác với CSDL..

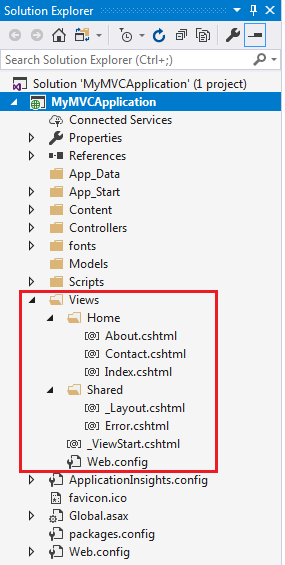
### ****3.7.**** ****Scripts****

Scripts : thư mục này chứa các tập tin JavaScript hay VBScript. MVC 5 chứa các tập tin mặc định :javascript  bootstrap, jquery 1.10.



### ****3.8.**** ****Views****

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



## 4. Controller

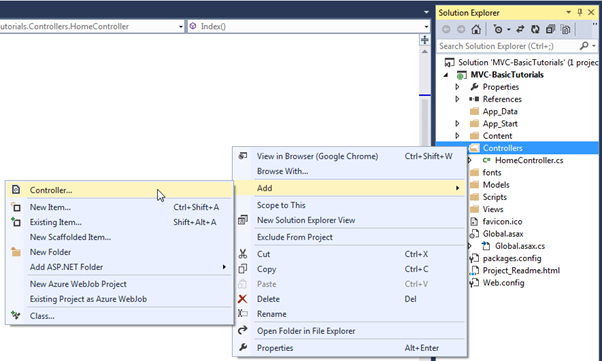
### 4.1. Trách nhiệm của Controller

Controller có 3 trách nhiệm chính:

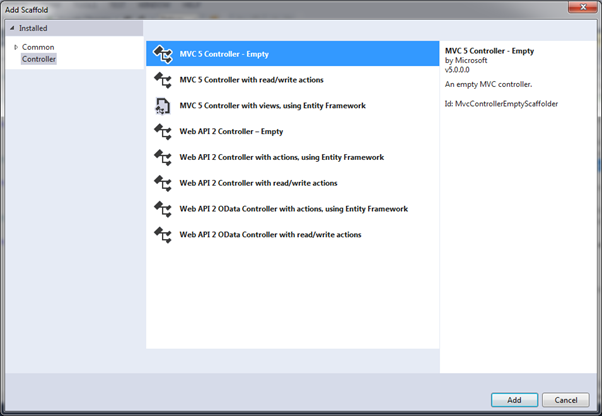
* Nhận request: Controller có trách nhiệm nhận request từ user.
* Dựng model: Controller Action method thực thi logic của ứng dụng và xây dựng nên model.
* Gửi trả response: Cuối cùng nó sẽ trả về kết quả trong HTML, File, JSON, XML hoặc bất cứ định dạng nào về cho user.

### 4.2. Thêm mới Controller

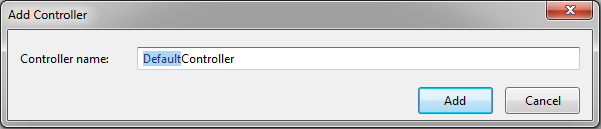
* Để thêm 1 controller mới, ở cửa sổ **Solution Explorer** của dự án **MVC-BasicTutorials**, chuột phải lên **thư mục Controllers** và sau đó nhấn **Add**, chọn **Controller**.



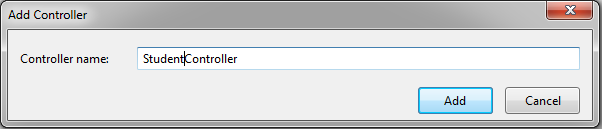
Ở hộp thoại **Add Scaffold**, chọn **MVC 5 Controller – Empty** và nhấn **Add**. Ở bước này, do chúng ta mới làm quen với Controller cho nên chúng ta sử dụng mẫu Controller dạng Empty, tức là nội dung trống. Từ đó, chúng ta sẽ viết 1 số đoạn code làm quen. Ở những bài tiếp theo, bạn sẽ làm quen các dạng Controller khác.

Scaffolding là một Framewrok tạo code tự động cho các ứng dụng web ASP.NET. Scaffolding giúp giảm thời gian phát triển **Controller**, **View**, v.v. trong Framework MVC. Bạn có thể phát triển một Scaffolding tùy ý bằng cách sử dụng các mẫu T4 theo tiêu chuẩn kiến trúc và code của bạn.  
 

Thêm Scaffold chứa các mẫu khác nhau để tạo bộ điều khiển mới. Chúng ta sẽ tìm hiểu về các mẫu khác sau. Hiện tại, chọn "**MVC 5 Controller - Empty**" và nhấp vào Thêm. Nó sẽ mở hộp thoại Thêm điều khiển như hình sau:



Sau đó, đặt tên Controller là **StudentController**, lưu ý bạn nên để tiền tố “Controller” cuối cùng khi đặt tên để dễ phân biệt lớp code nào là Controller.



Ở bước này, thư mục Controller, lớp **StudentController.cs** được tạo ra với nội dung mặc định như trong đoạn code sau.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace MVC\_BasicTutorials.Controllers

{

public class StudentController : Controller

{

// GET: Student

public ActionResult Index()

{

return View();

}

}

Như bạn có thể thấy ở trên, lớp StudentContoder kế thừa từ lớp **Controller**. Mỗi controller trong MVC phải kế thừa từ lớp **Controller**trừu tượng này. Lớp **Controller**cơ sở này chứa các phương thức có thể được sử dụng cho các mục đích khác nhau.

Tiếp đến, bạn thêm 1 số đoạn mã vào tập tin **StudentController.cs** như sau:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace MVC\_BasicTutorials.Controllers

{

public class StudentController : Controller

{

// GET: Student

public string Index()

{

return "This is Index action method of StudentController";

}

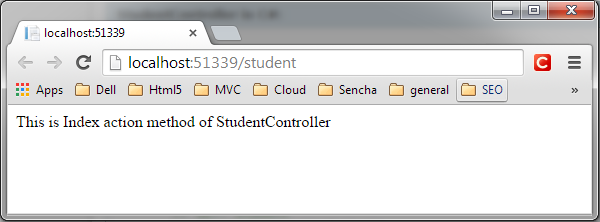
}

}

Giải thích đoạn mã trên:

Phương thức Index() trả về kiểu string với giá trị là “This is Index action method of StudentController.” Đây là phương thức mặc định của 1 Controller bất kỳ.

Bạn có thể tạo nhiều phương thức thực thi ở tập tin **StudentController**.cs tùy ý. Tiếp theo, bạn thực thi ứng dụng **bằng cách nhấn F5 hoặc Ctrl + F5** (chế độ không cần Debug) để xem kết quả.

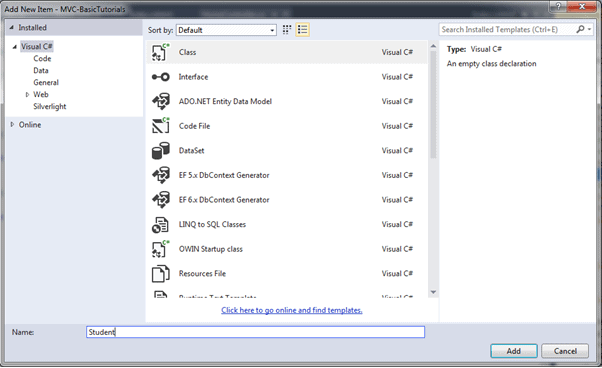


## 5. Models

### ****Cách thêm một Model trong ASP.NET MVC****

Quay trở lại dự án trước, trong cửa sổ Solution Explorer, bạn có thể thấy một thư mục tên là Model. Chuột phải lên thư mục này, chọn Add - -> click on Class...

Trong hộp thoại Add New Item, nhập tên lớp 'Student' và click **Add**.



Đã thêm một lớp Student trong thư mục model. B6y giờ, chúng ta thêm các thuộc tính Id, Name, Age.

namespace MVC\_BasicTutorials.Models

{

public class Student

{

public int StudentId { get; set; }

public string StudentName { get; set; }

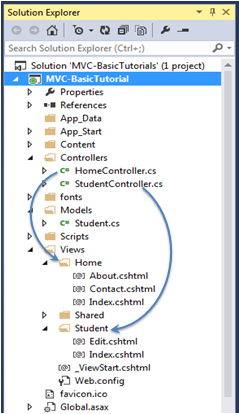
public int Age { get; set; }

}

}

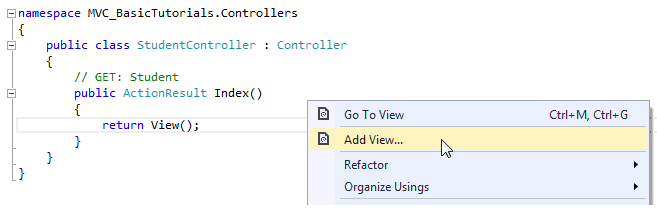
## 6. Views

View  ASP.NET MVC được lưu trong thư mục Views. Các phương thức hành động khác nhau của một lớp controller duy nhất có thể hiển thị các **Views** khác nhau, do đó, thư mục **Views** chứa một thư mục riêng cho mỗi controller có cùng tên controller, để phù hợp với nhiều **Views**.Ví dụ: các **Views** sẽ được hiển thị từ bất kỳ phương thức hành động nào của HomeController, nằm trong thư mục **Views > Home.** Tương tự các Views sẽ được hiển thị từ StudentController, sẽ nằm trong thư mục **Views > Student**như được hiển thị bên dưới.

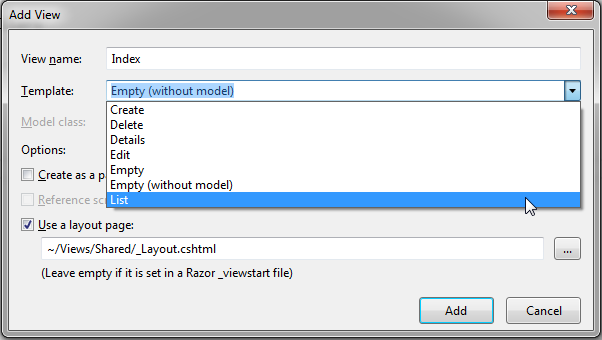


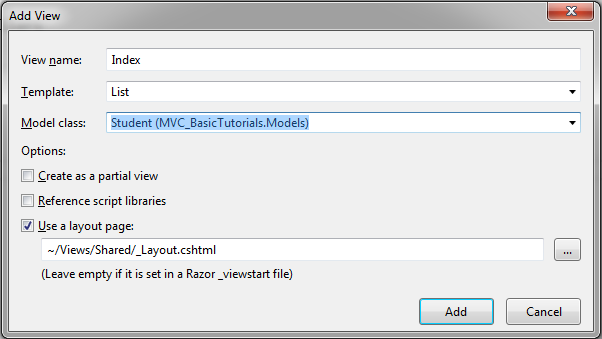
**Tạo mới View.**

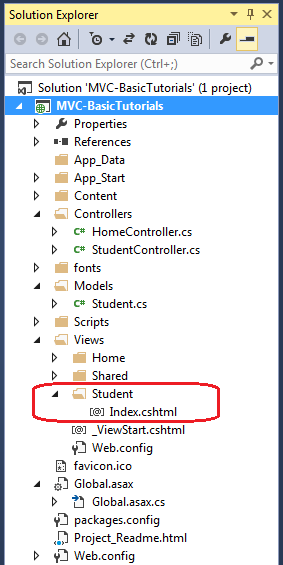
Click chuột phải vào ActionResult Index(), chọn Add View…



Trong cửa sổ Add View, các bạn nhập các thông số, sau đó nhấn nút Add để tiến hành tạo View.







Views\Student\Index.cshtml:

@model IEnumerable<MVC\_BasicTutorials.Models.Student>

@{

ViewBag.Title = "Index";

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

}

<h2>Index</h2>

<p>

@Html.ActionLink("Create New", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.StudentName)

</th>

<th>

@Html.DisplayNameFor(model => model.Age)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.StudentName)

</td>

<td>

@Html.DisplayFor(modelItem => item.Age)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id=item.StudentId }) |

@Html.ActionLink("Details", "Details", new { id=item.StudentId }) |

@Html.ActionLink("Delete", "Delete", new { id = item.StudentId })

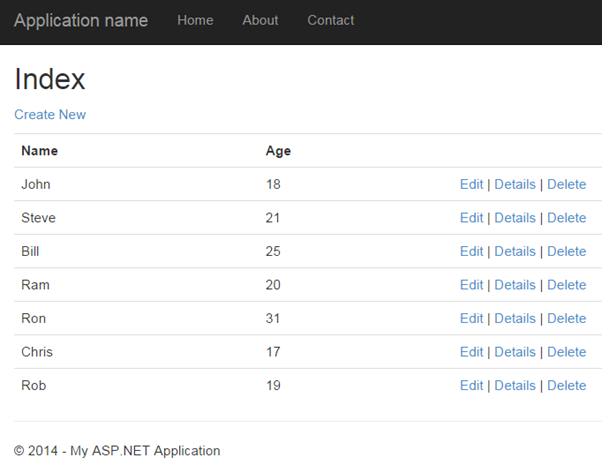
</td>

</tr>

}

</table>

Run chúng ta có kết quả:



## 7. Phương thức Action

**Phương thức Action mặc định:**

routes.MapRoute(

name: "Default",

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

defaults: new { controller = "Home",

action = "Index",

id = UrlParameter.Optional

});

**Phương thức ActionResult:**

|  |  |
| --- | --- |
| **Result Class** | **Description** |
| ViewResult | Trả về HTML |
| EmptyResult | Không trả về cái gì cả. |
| ContentResult | Trả về chuỗi ký tự |
| JavaScriptResult | Trả về  JavaScript script. |
| JsonResult | Trả về một JSON có thể sử dụng trong AJAX |
| RedirectResult | Chuyển hướng người dùng |
| RedirectToRouteResult | Chuyển hướng hành động khác. |
| PartialViewResult | Trả về HTML từ Partial view |
| Trả về HTML từ Partial view | Trả về lỗi và HTTP Code |

## 8. Tích hợp Controller , Models và Views

**StudentController:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using MVC\_BasicTutorials.Models;

namespace MVC\_BasicTutorials.Controllers

{

public class StudentController : Controller

{

// GET: Student

public ActionResult Index()

{

return View();

}

}

}

**Lớp Model Student**

namespace MVC\_BasicTutorials.Models

{

public class Student

{

public int StudentId { get; set; }

public string StudentName { get; set; }

public int Age { get; set; }

}

}

**Tập tin Index.cshtml để hiển thị danh sách sinh viên**

@model IEnumerable<MVC\_BasicTutorials.Models.Student>

@{

ViewBag.Title = "Index";

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

}

<h2>Index</h2>

<p>

@Html.ActionLink("Create New", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.StudentName)

</th>

<th>

@Html.DisplayNameFor(model => model.Age)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.StudentName)

</td>

<td>

@Html.DisplayFor(modelItem => item.Age)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id=item.StudentId }) |

@Html.ActionLink("Details", "Details", new { id=item.StudentId }) |

@Html.ActionLink("Delete", "Delete", new { id = item.StudentId })

</td>

</tr>

}

Bây giờ, chúng ta cần truyền dữ liệu trong Model từ Controller sang View. Chúng ta sử dụng IEnumerable để lưu trữ các đối tượng sinh viên như sau:

public class StudentController : Controller

{

// GET: Student

public ActionResult Index()

{

var studentList = new List<Student>{

new Student() { StudentId = 1, StudentName = "John", Age = 18 } ,

new Student() { StudentId = 2, StudentName = "Steve", Age = 21 } ,

new Student() { StudentId = 3, StudentName = "Bill", Age = 25 } ,

new Student() { StudentId = 4, StudentName = "Ram" , Age = 20 } ,

new Student() { StudentId = 5, StudentName = "Ron" , Age = 31 } ,

new Student() { StudentId = 4, StudentName = "Chris" , Age = 17 } ,

new Student() { StudentId = 4, StudentName = "Rob" , Age = 19 }

};

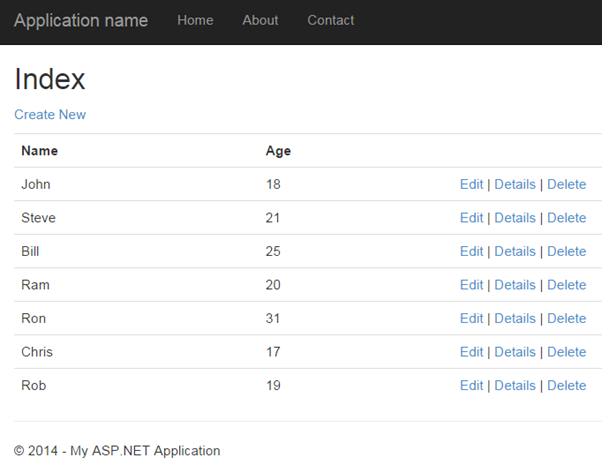
// Get the students from the database in the real application

return View(studentList);

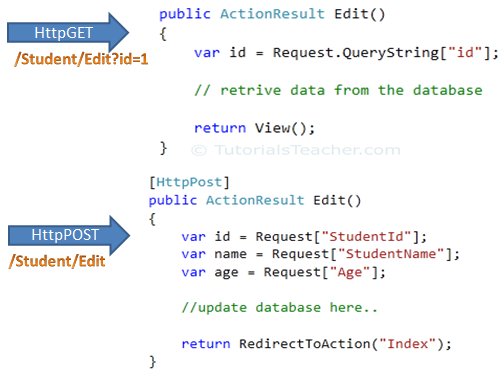
}

}

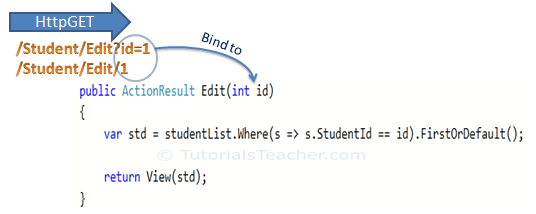
Bây giờ, bạn có thể chạy project MVC bằng cách nhấn F5 cho kết quả:

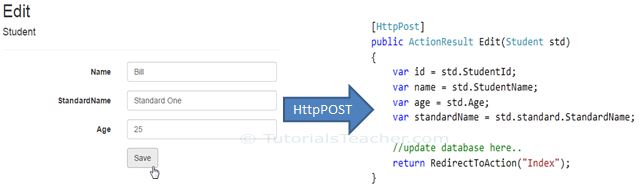


## 9. Model Binding

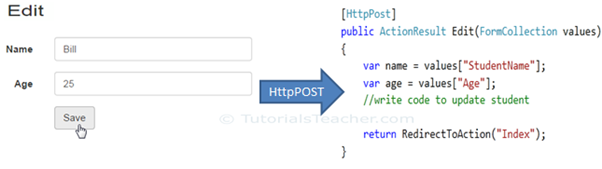


**Liên kết với kiểu nguyên thủy**

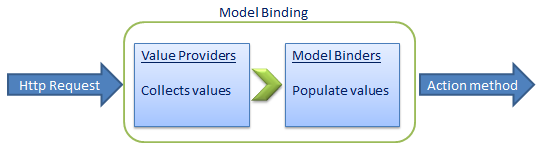




**Sử dụng FormCollection để gửi dữ liệu từ View lên Action**



**Liên kết mô hình bên trong**



# Chương II: PHÂN TÍCH DỰ ÁN

**1. Các biên bản họp nhóm**

**BIÊN BẢN HỌP**

Về việc thống nhất nội dung và triển khai dự án

Hôm nay, lúc 09 giờ ngày 2 tháng 3 năm 2020 tại phòng E1-207 trường Đại học Thủ Dầu Một diễn ra cuộc họp với các nội dung sau:

**I. Thành phần tham dự:**

* Chủ toạ: Tăng Trung Hiếu
* Thư ký: Lê Văn Đông.

**II. Nội dung cuộc họp:**

1. Tiến hành chọn và thống nhất đề tài của dự án:

1. **Tên đề tài:** Xây dựng Website Tải Game miễn phí
2. **Địa điểm thực hiện:** Địa bàn thành phố Thủ Dầu Một
3. **Trưởng nhóm dự án:** Tăng Trung Hiếu
4. **Đội phát triển dự án:**

* Tăng Trung Hiếu
* Lê Văn Đông

**2. Phân công nhiệm vụ**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | Chức vụ | Nhiệm vụ |
| 1 | Tăng Trung Hiếu | Trưởng dự án | Tiếp nhận dự án, dự trù nhân lực.  Gặp gỡ khách hàng.  Hoàn thành bản kế hoạch tổng thể.  Định hướng và phân công công việc trong quá trình triển khai thực hiện dự án.  Xây dựng và thực thi hệ thống.  Thiết kế mô hình và lập trình cơ sở dữ liệu. |
| 2 | Lê Văn Đông | Phó dự án | Gặp mặt các chuyên gia tư vấn trong lĩnh vực.  Đưa ra các gợi ý trong việc xây dựng phần mềm. Đưa ra các quyết định trong việc xây dựng giao diện cảm nhận dựa trên yêu cầu của khách hàng tương ứng.  Kiểm thử hệ thống.  Xem xét các rủi ro gặp phải.  Phân tích thiết kế, thực thi hệ thống.  Kiểm tra chất lượng công việc trong quá trình thực hiện dự án và đưa ra báo cáo về cho trưởng dự án.  Triển khai hệ thống tới khách hàng. |

Cuộc họp kết thúc lúc 10 giờ 30 phút cùng ngày.

**Thư ký cuộc họp** **Chủ trì cuộc họp**

*(Đã ký) (Đã ký)*

**Lê Văn Đông**   **Tăng Trung Hiếu**

**2. Danh sách các chức năng đề tài cần đáp ứng**

|  |  |  |  |
| --- | --- | --- | --- |
| ***STT*** | ***TÊN CHỨC NĂNG CẤP 1*** | ***TÊN CHỨC NĂNG CẤP 2*** | ***TÊN CHỨC NĂNG CẤP 3*** |
| ***NHỮNG CHỨC NĂNG PHẢI LÀM*** | | |
| ***1*** | Quản lý game | Thêm game |  |
| Tìm kiếm game |  |
| Cập nhật thông tin game |  |
| Xóa thông tin game |  |
| ***2*** | Quản lý khách hàng | Chỉnh sửa thông tin khách hàng |  |
| Trả lời phản hồi |  |
| Xóa thông tin khách hàng |  |
| Hiển thị lịch sử tải game |  |
| ***3*** | Quản lý menu | Thêm menu |  |
| Sửa menu |  |
| Xóa đơn hàng |  |
| ***4*** | Quản lý tài khoản | Quản trị viên | Xem thông tin cá nhân nhân viên |
|  |  | Thêm thông tin cá nhân nhân viên |
|  |  | Sửa thông tin cá nhân nhân viên |
|  |  | Xóa thông tin cá nhân nhân viên |
|  |  | Nhân viên | Xem thông tin cá nhân |
|  |  |  | Sửa thông tin cá nhân |

## 3. Danh sách chức năng đề tài không làm

|  |  |  |  |
| --- | --- | --- | --- |
| ***STT*** | ***TÊN CHỨC NĂNG CẤP 1*** | ***TÊN CHỨC NĂNG CẤP 2*** | ***GHI CHÚ*** |
| ***NHỮNG CHỨC NĂNG KHÔNG LÀM*** | | |
| 1 | Quản lý phản hồi | Đánh giá sao |  |

**4. Công cụ để làm đề tài**

- Sử dụng SQL Server làm cơ sở dữ liệu thời gian thực

- Visual Studio dùng để thiết kế giao diện và code các chức năng.

**5. Cấu hình tối thiểu để chạy được sản phẩm của đề tài**

**-** Sản phẩm chạy trên các trình duyệt web( IE8 , Chrome, Firefox,… ).

**6. Các biểu đồ Use Case của đề tài**



Hình 1. Sơ đồ Use Case Chức năng chính



Hình 2. Sơ đồ Use Case Quản lý Game



Hình 3. Sơ đồ Use Case Quản lý tài khoản



Hình 4. Sơ đồ Use Case Quản lý Tải Game

# Chương III: Thiết kế cơ sở dữ liệu

### 3.1 Bảng dữ liệu

### 3.1.1 Bảng dữ liệu MenuType

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MenuType** | | | | |
| **Tên cột** | **Kiểu dữ liệu cho cột** | **Cho phép NULL** | **Khóa** | **Mô tả thêm** |
| ID | Int | Not null | PK | Mã Menu |
| Name | Nvarchar(50) | null |  | Tên Menu |

-Bảng này dùng để miêu tả cấp menu

**3.1.2 Bảng dữ liệu menu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Menu** | | | | |
| **Tên cột** | **Kiểu dữ liệu cho cột** | **Cho phép NULL** | **Khóa** | **Mô tả thêm** |
| ID | int | Not null | PK | Khóa Chính |
| Text | Nvarchar(150) | null |  | Tên Menu |
| Link | Nvarchar(250) | null |  | Liên Kết |
| Target | Varchar(50) | null |  |  |
| TypeID | Int | Null | FK | Khóa Ngoại |
| Status | Bit | null |  | Trạng thái |
| Submenu | Int | Null |  | Menu Phụ |
| Multilevel | Bit | Null |  | Cấp Menu |

-Bảng này dùng để lưu trữ dữ liệu của menu

### 3.1.3 Bảng dữ liệu GameCategory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GameCategory** | | | | |
| **Tên cột** | **Kiểu dữ liệu cho cột** | **Cho phép NULL** | **Khóa** | **Mô tả thêm** |
| ID | int | Not null | PK | Khóa Chính |
| Name | Nvarchar(50) | null |  | Tên Danh Mục |
| UnTitle | Nvarchar(50) | Null |  |  |
| SeoTitle | Nvarchar(50) | null |  |  |
| ShowOnhome | Bit | null |  |  |

-Bảng này dung để lưu trữ thể loại game

**3.1.4 Bảng dữ liệu Game**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Game** | | | | |
| **Tên cột** | **Kiểu dữ liệu cho cột** | **Cho phép NULL** | **Khóa** | **Mô tả thêm** |
| ID | int | Not null | PK | Khóa Chính |
| Name | Nvarchar(100) | null |  | Tên Game |
| UnTitle | Nvarchar(50) | null |  |  |
| IDCaterogy | Int | null |  | Danh Mục |
| Code | Nvarchar(250) | null |  |  |
| Description | Nvarchar(1000) | Null |  | Mô Tả Game |
| Image | Nvarchar(250) | Null |  | Hình Game |
| MoreImage | Xml | Null |  | Hình thêm |
| Quality | Int | Null |  | Số lượng |
| Detail | Ntext | Null |  | Chi Tiết Game |
| ViewCount | Int | Null |  | Lượt xem |

-Bảng này dùng để lưu trữ dữ liệu game

### 3.1.5 Bảng dữ liệu TaiKhoan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TaiKhoan** | | | | |
| **Tên cột** | **Kiểu dữ liệu cho cột** | **Cho phép NULL** | **Khóa** | **Mô tả thêm** |
| ID | int | Not null | PK | Khóa Chính |
| Username | Nvarchar(50) | null |  | Tên đăng nhập admin |
| Displayname | Nvarchar(150) | Null |  | Tên Admin |
| Gender | Nvarchar(10) | null |  | Giới Tính |
| Phone | Varchar(15) | null |  | Số điện thoại admin |
| Address | Nvarchar(500) | Null |  | Địa chỉ Admin |
| Ngaysinh | datetime | Null |  | Ngày Sinh Admin |
| Image | Nvarchar(250) | Null |  | Hình Admin |
| TypeAccount | Varchar(10) | Null |  | Loại Tài Khoản |

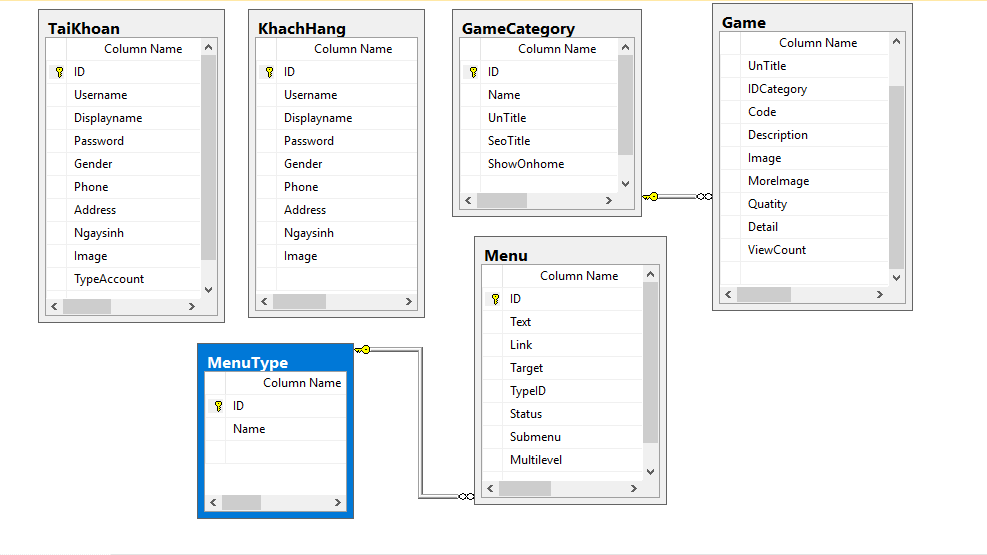
-Bảng này dùng để lưu trữ dữ liệu tài khoản quản trị viên

**3.1.6 Bảng dữ liệu KhachHang**

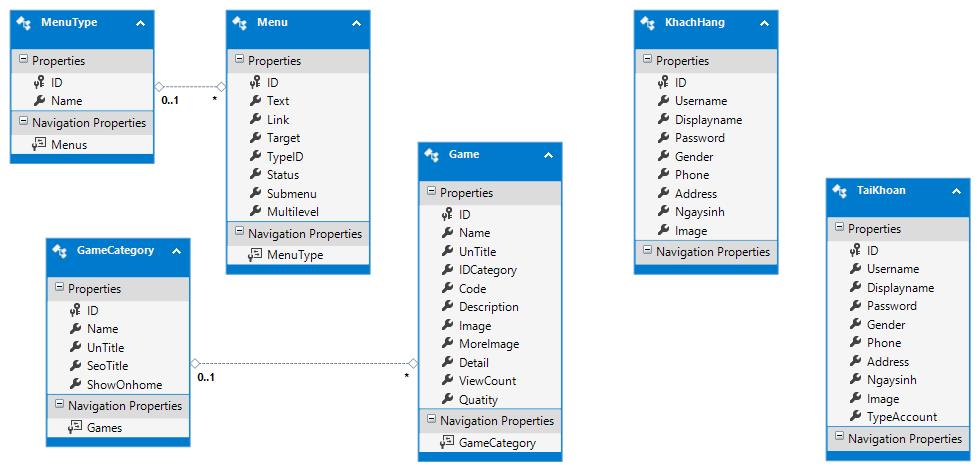
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KhachHang** | | | | |
| **Tên cột** | **Kiểu dữ liệu cho cột** | **Cho phép NULL** | **Khóa** | **Mô tả thêm** |
| ID | int | Not null | PK | Khóa Chính |
| Username | Nvarchar(50) | null |  | Tên đăng nhập Khách hàng |
| Displayname | Nvarchar(150) | Null |  | Tên Khách hàng |
| Gender | Nvarchar(10) | null |  | Giới Tính |
| Phone | Varchar(15) | null |  | Số điện thoại Khách hàng |
| Address | Nvarchar(500) | Null |  | Địa chỉ Khách hàng |
| Ngaysinh | datetime | Null |  | Ngày Sinh Khách hàng |
| Image | Nvarchar(250) | Null |  | Hình Khách hàng |
| Password | Varchar(250) | Null |  | Mật khẩu Khách hàng |

-Bảng này dùng để lưu dữ liệu tài khoản khách hang

## 3.2 Diagram



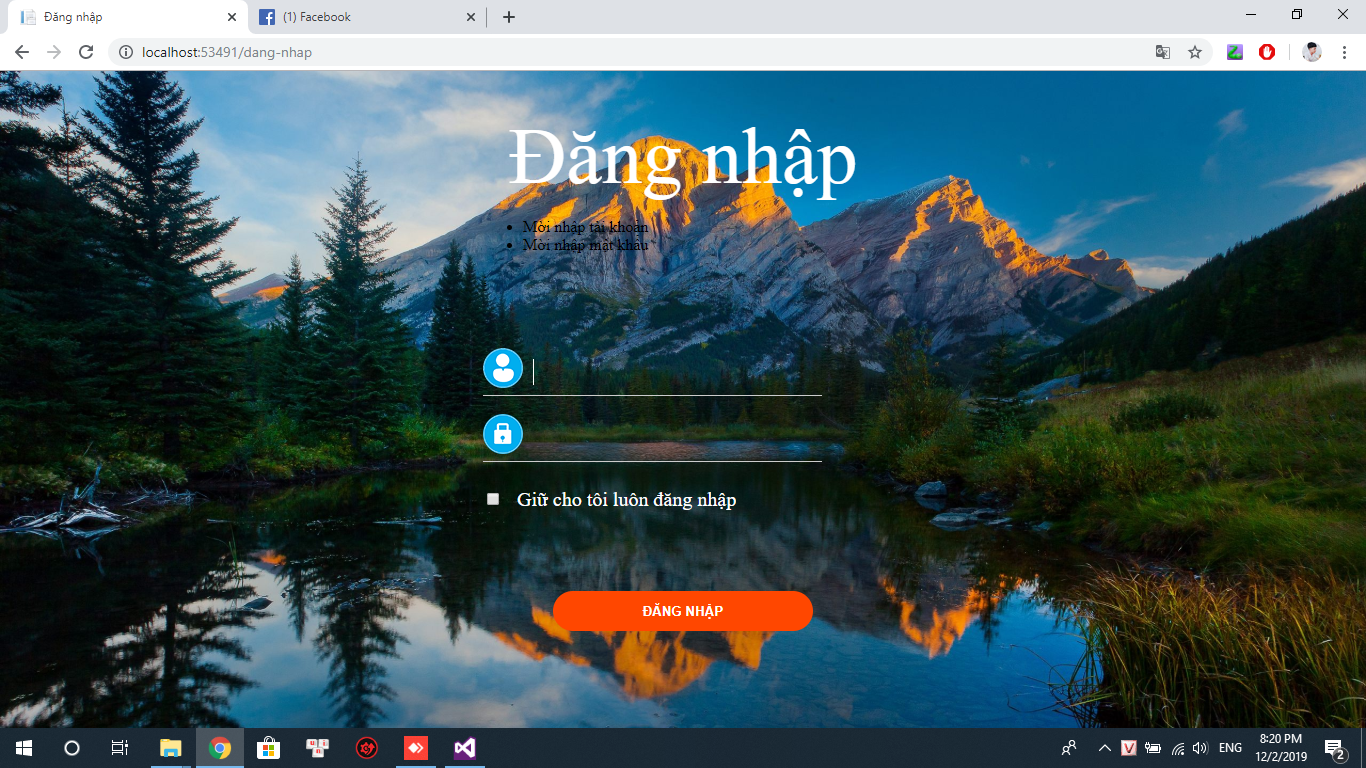
## 3.3 Biểu đồ lớp



# Chương IV: TRIỂN KHAI , ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

**1. Các chức năng đã làm được**

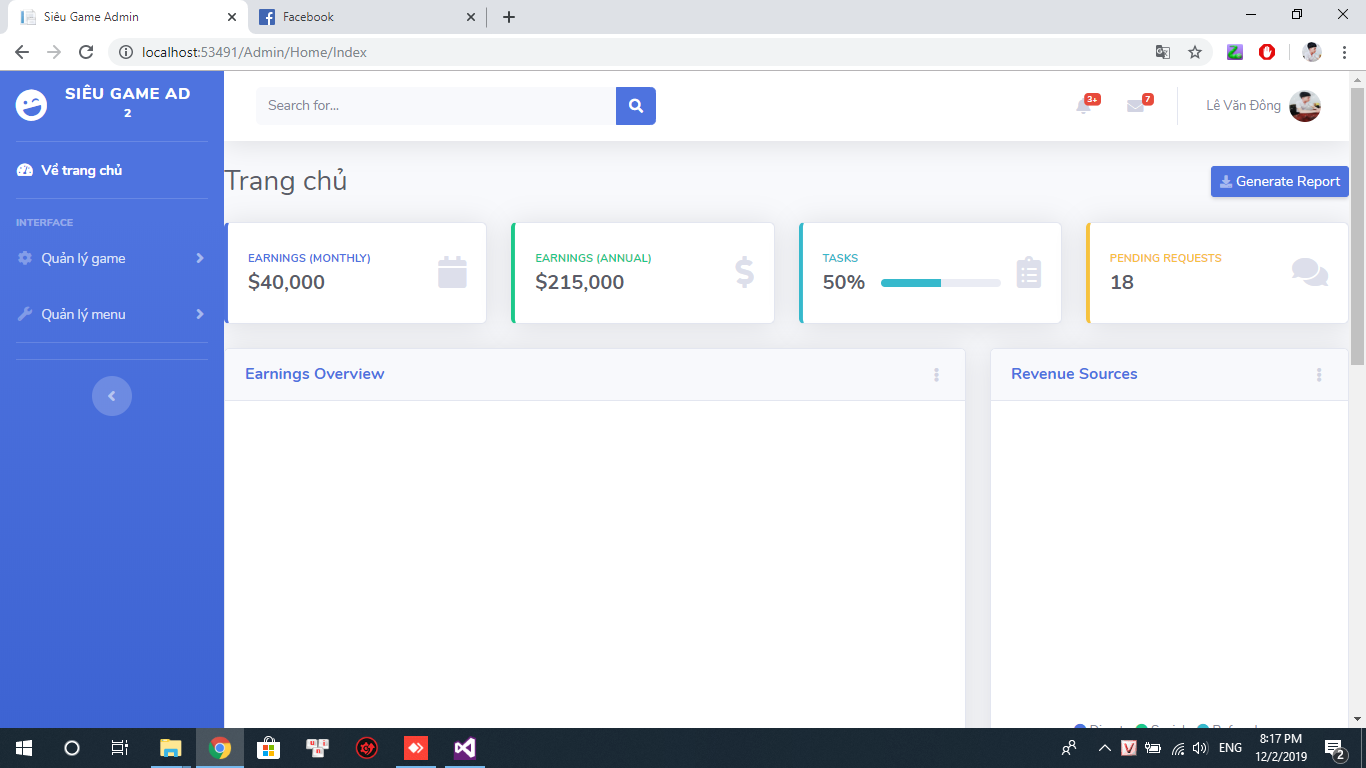
### 1.1. Chức năng đăng nhập



Hình 3.1.1 Giao diện đăng nhập

## 2. Chức năng chính

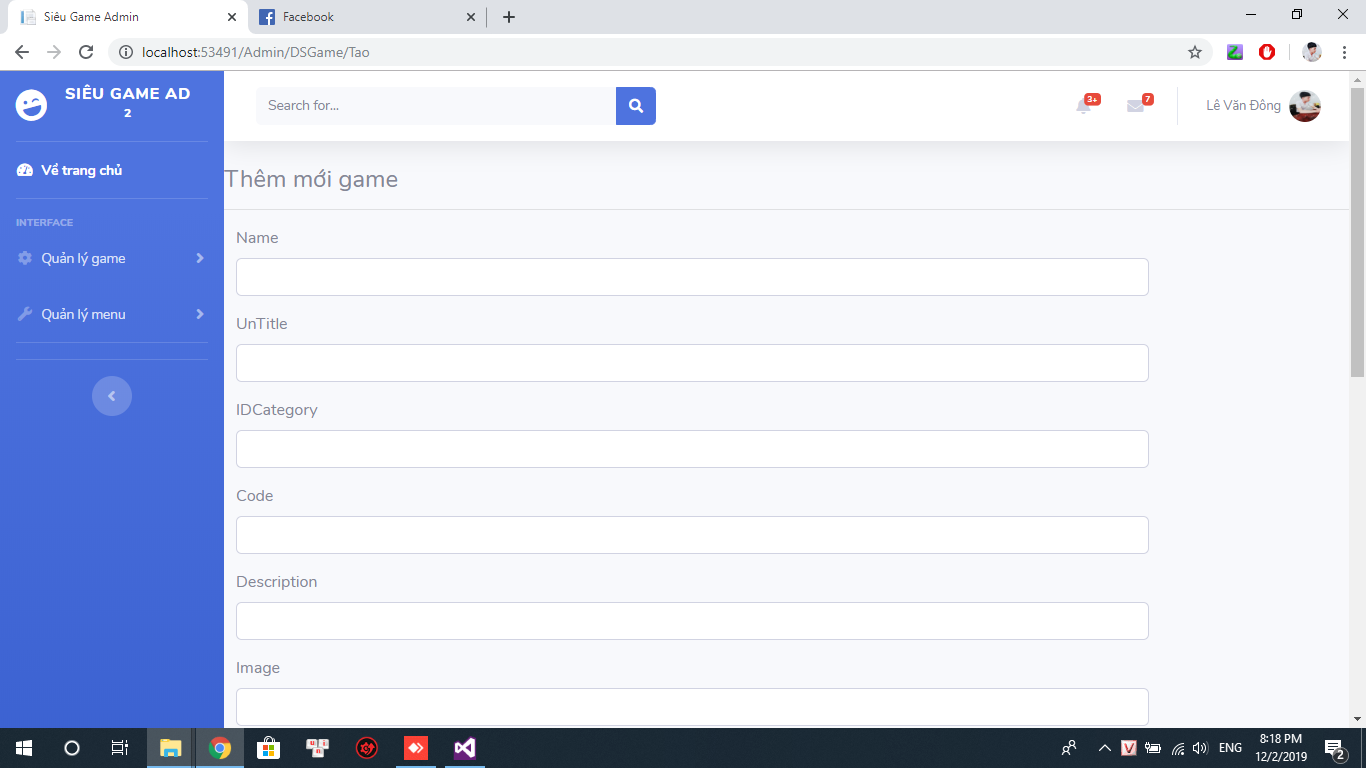
### 2.1. Giao diện dành cho người quản trị



* Đây là giao diện chính của phần mềm với các chức năng như sau:
* Quản Lý Game
* Quản Lý Menu

Hình 3.1.2 Giao diện quản trị

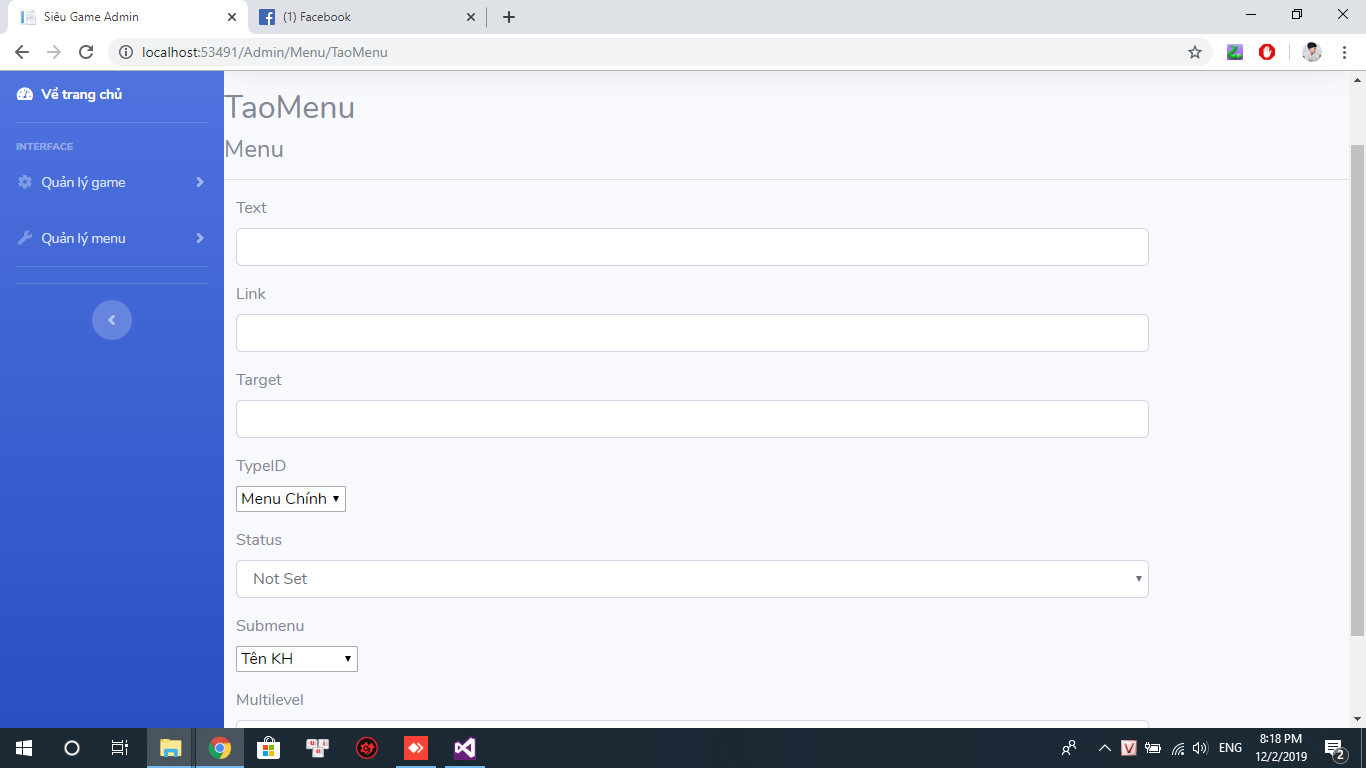
### 2.2. Giao diện Quản Lý Game



* Giao diện quản lý game có các chức năng:
  + - +Thêm mới game
    - +Sửa thông tin game
    - +Xóa Game

Hình 3.1.3 Giao diện chức năng Quản lý Game

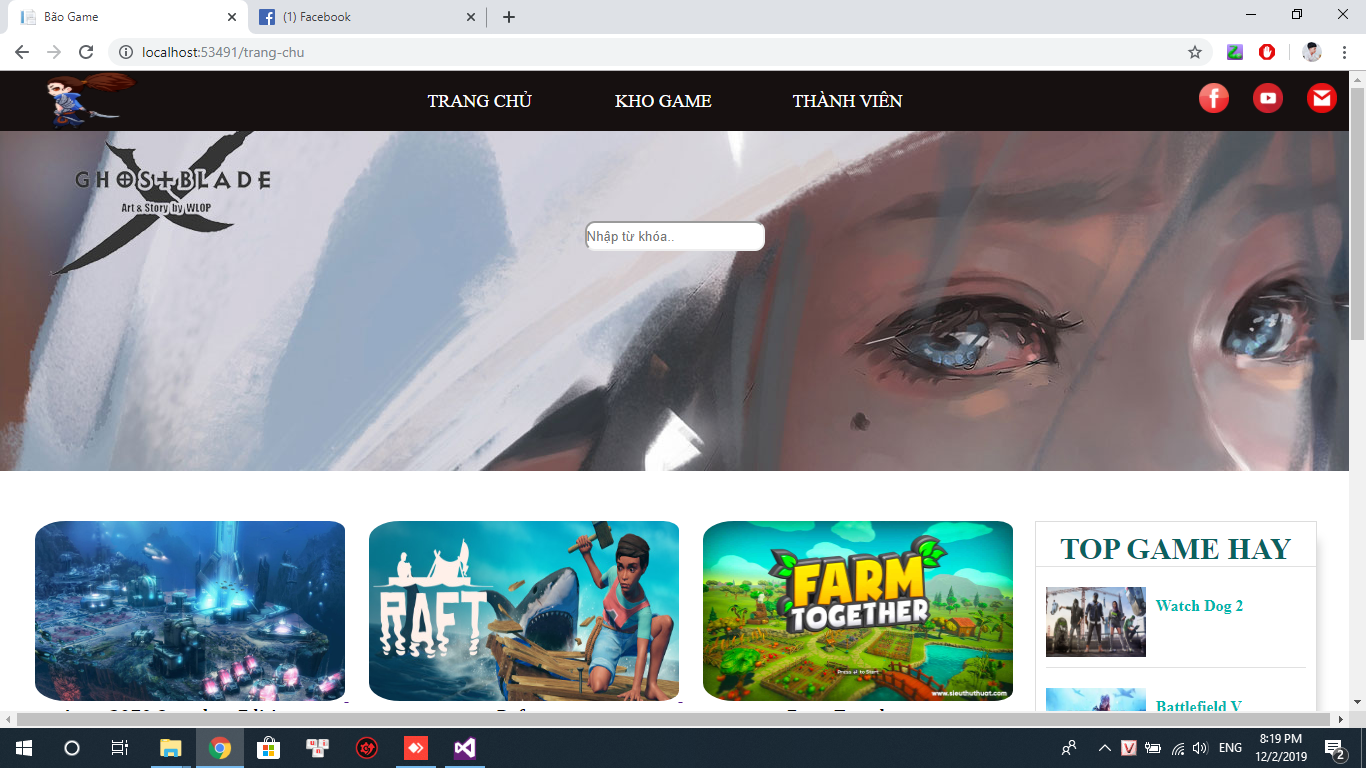
### 2.3. Giao diện Quản Lý Menu

****

* Giao diện quản lý menu có các chức năng:
* +Thêm mới menu
* +Sửa menu
* +Xóa menu

Hình 3.1.4 Giao diện quản lý Menu

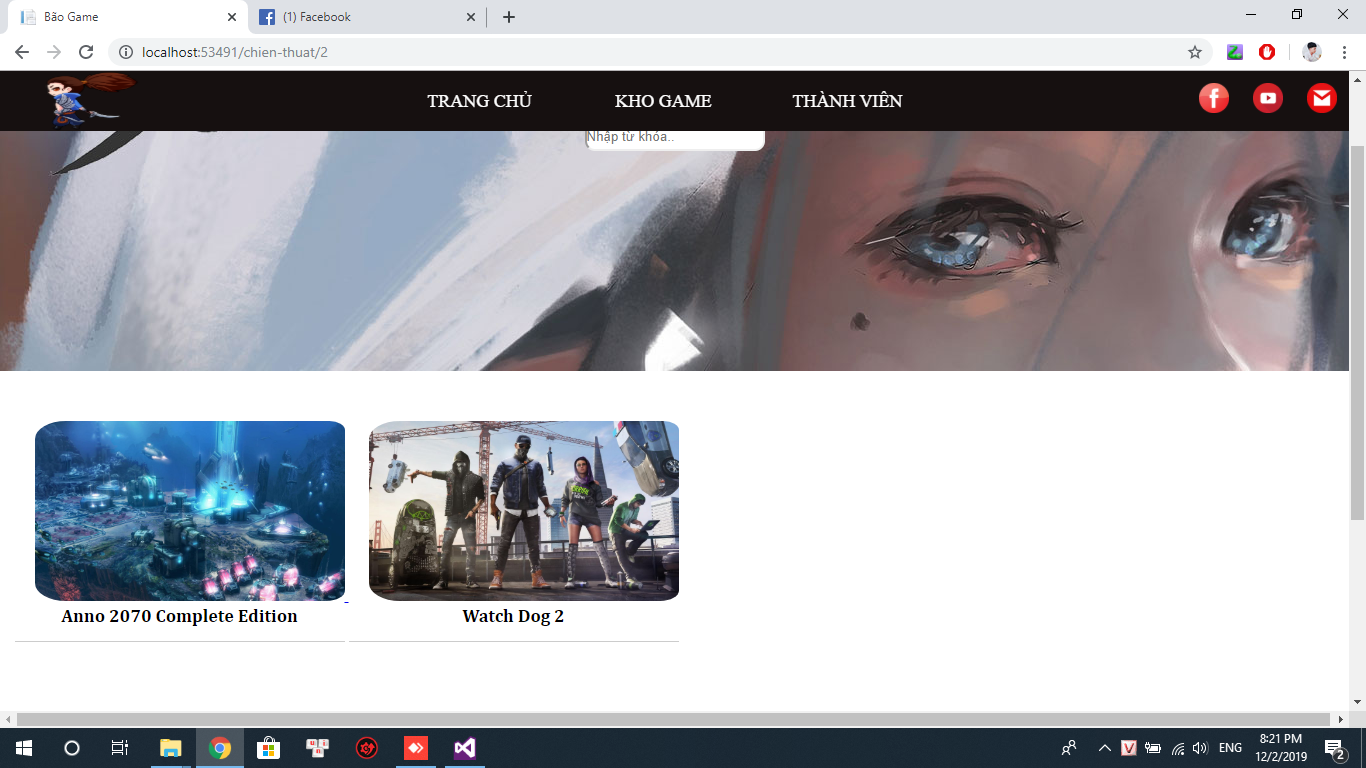
### 2.4. Giao diện Dành cho khách hàng

****

* Khách hàng có thể xem được cái game hay và tải về theo ý muốn

Hình 3.1.5 Giao diện dành cho khách hàng

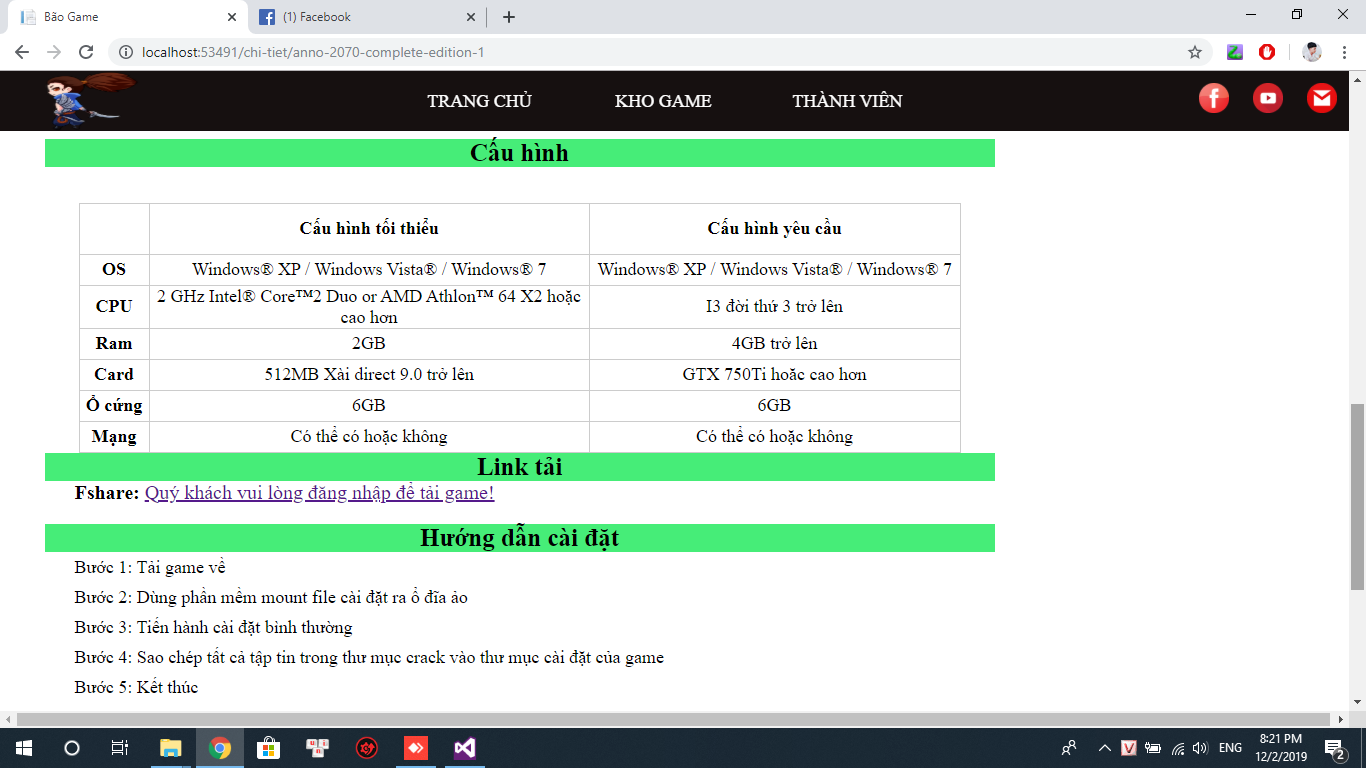
### 2.5. Giao diện Game theo thể loại



Hình 3.1.6 Giao diện Game theo thể loại

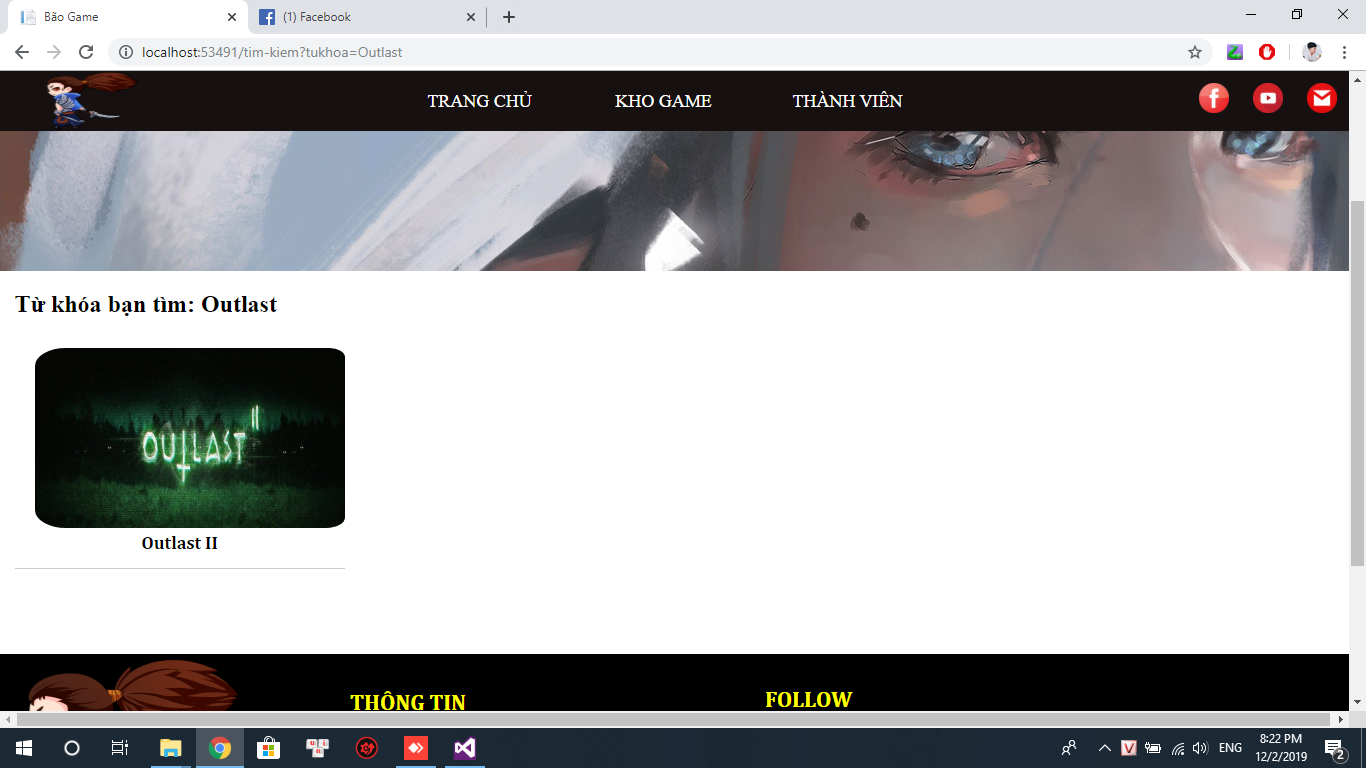
### 2.6. Giao diện Chi tiết Game





Hình 3.1.7 Giao diện chi tiết Game

### 2.7. Giao diện tìm kiếm Game



## 2. Những khó khăn và hạn chế trong quá trình thực hiện đề tài

* Chưa có kiến thức về phân tích thiết kế hệ thống dẫn đến việc khó khăn xây dựng website.
* Tài liệu tham khảo còn hạn chế
* Phân bố thời gian lập trình không hợp lý , bỏ qua 1 số giai đoạn.
* Chưa có kinh nghiệm trong việc viết và thiết kế website thực tế dẫn đến tốn nhiều thời gian để tìm hiểu.

## 3. Hướng phát triển của đề tài

* Cập nhật thêm các chức năng như ưu đãi , phân công , quản lý kho game …
* Tối ưu hóa cơ sở dữ liệu.
* Nâng cấp website

# TÀI LIỆU THAM KHẢO

1. **Khóa học xây dựng website MVC áp dụng vào dự án thực tế**

**(**<https://www.youtube.com/watch?v=M0jdFS4ZyEk&list=PLRhlTlpDUWsyK1TIsewrQ7WwC7QkCSCPD>)

1. **Tài liệu về MVC**

**(**<https://cuongquach.com/giao-trinh-asp-net-mvc-5-nhat-nghe-pdf.html>)