

Trường Đại Học Công Nghệ Sài Gòn
Khoa Công Nghệ Thông Tin
---oOo---



Giáo trình thực hành:

LẬP TRÌNH WINDOWS

(Lưu hành nội bộ)

Năm 2020

I. MỤC TIÊU:

Trong bài thực hành này, sinh viên sẽ học:

- Thẻ input, form, table.
- Đối tượng Request, ViewBag.
- Thao tác trên 1 Table trong CSDL
- Thực hiện các phép toán hiển thị danh sách, thêm, xóa, sửa.

II. TÓM TẮT LÝ THUYẾT:**1/ Thẻ input, form, table:****a) Thẻ input**

- ❖ **Miêu tả:** thẻ HTML <input> được sử dụng để khai báo phần tử input - một điều khiển mà cho phép người sử dụng nhập dữ liệu đầu vào. Thuộc tính **type** có thể xác định kiểu điều khiển, có thể có các giá trị : button, checkbox, color, date, datetime, datetime-local, email, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week.

Ví dụ:

```
<html>
<head>
  <title>Vi du the input trong HTML</title>
</head>
<body>
  <form action="/cgi-bin/hello_get.cgi" method="get">
    First name:
    <input type="text" name="first_name" value="" maxlength="100" />
    <br />
    Last name:
    <input type="text" name="last_name" value="" maxlength="100" />
    <input type="submit" value="Submit" />
  </form>
</body>
</html>
```

Sẽ cho kết quả :

First name:

Last name:

❖ **Một số thuộc tính thường dùng:**

Thuộc tính	Giá trị	Miêu tả
id	Chuỗi	Mỗi phần tử có một id duy nhất
maxlength	Số	Xác định số ký tự tối đa được cho phép trong trường văn

		bản
min	Số	Xác định giá trị nhỏ nhất
max	Số	Xác định giá trị lớn nhất
multiple	multiple	Xác định rằng người sử dụng có thể nhập nhiều giá trị
name	chuỗi	Tên của điều khiển
readonly	readonly	Thiết lập điều khiển đầu vào là chỉ đọc (read-only)
value	chuỗi	Xác định giá trị khởi tạo cho điều khiển

b) Thẻ form

- ❖ **Miêu tả:** Form trong HTML thường được sử dụng để thu thập dữ liệu đầu vào của người sử dụng. Phần tử **<form>** định nghĩa một Form trong HTML:

```
1 <form action="server url" method="get|post">
2 //input controls ví dụ: textfield, textarea, radiobutton, button
3 </form>
```

- ❖ **Một số thuộc tính thường dùng:**

Thuộc tính	Miêu tả
action	Ứng dụng quản trị back-end sẵn sàng để xử lý dữ liệu từ site khách.
method	Phương thức để tải dữ liệu lên. Thường sử dụng là GET và POST.

c) Thẻ table

- ❖ **Miêu tả:** Thẻ HTML **<table>** được sử dụng để định nghĩa một bảng. Thẻ table chứa các thẻ khác để định nghĩa cấu trúc của một bảng, trong đó thẻ **<tr>** được sử dụng để tạo các hàng và thẻ **<td>** được sử dụng để tạo các ô.

Ví dụ:

```
<html>
<head>
  <title>Vi du bang trong HTML</title>
</head>
<body>
  <table border="1">
    <tr>
      <td>Row 1, Column 1</td>
      <td>Row 1, Column 2</td>
    </tr>
    <tr>
      <td>Row 2, Column 1</td>
      <td>Row 2, Column 2</td>
    </tr>
  </table>
</body>
</html>
```

Sẽ cho kết quả :

Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

2/ Đối tượng Request

- ❖ Công dụng: Đối tượng Request được dùng để nhận những thông tin từ trình duyệt của người dùng gửi về cho Web Server. Những thông tin này gồm các thông số của Form khi được Submit dùng phương thức POST hoặc GET hay các tham số được ghi cùng với trang ASP.NET trong lời gọi đến trang đó. Request object đại diện cho Client khi yêu cầu trang Web, còn Server sẽ dùng vừa Response vừa Request để đáp ứng yêu cầu hay đòi hỏi thông tin từ Client.

- ❖ Ví dụ:

```
Response.Write(Request.Form["ten"]);
```

Hoặc

```
Response.Write(Request["ten"]);
```

Hoặc:

```
Response.Write(Request.QueryString["ten"]);
```

3/ Đối tượng ViewBag

- ViewBag dùng để truyền dữ liệu tạm thời từ controller đến view, , dữ liệu tạm thời không chứa trong model.
- ViewBag là một thuộc tính động tận dụng các tính năng động mới trong C # 4.0
- Bạn có thể gán bất kỳ số lượng thuộc tính và giá trị nào cho ViewBag. Nếu bạn gán cùng tên thuộc tính nhiều lần cho ViewBag, thì nó sẽ chỉ xem xét giá trị cuối cùng được gán cho thuộc tính.
- Vòng đời của ViewBag trong suốt một request hiện hành. Giá trị của ViewBag sẽ là null nếu chuyển hướng xảy ra.

Ví dụ:

```
namespace MVC_BasicTutorials.Controllers
{
    public class StudentController : Controller
    {
        IList<Student> studentList = new List<Student>() {
            new Student(){ StudentID=1, StudentName="Steve", Age = 21 },
            new Student(){ StudentID=2, StudentName="Bill", Age = 25 },
            new Student(){ StudentID=3, StudentName="Ram", Age = 20 },
            new Student(){ StudentID=4, StudentName="Ron", Age = 31 },
            new Student(){ StudentID=5, StudentName="Rob", Age = 19 }
        };

        // GET: Student
        public ActionResult Index()
```

```
{
    ViewBag.TotalStudents = studentList.Count();

    return View();
}

}
```

Trong ví dụ trên, chúng ta muốn hiển thị tổng số sinh viên trong view. Vì vậy, đã gán số lượng sinh viên bằng `ViewBag.TotalStudents = studentList.Count()`. Bây giờ, trong view `Index.cshtml`, ta có thể truy cập thuộc tính `ViewBag.TotalStudents` và hiển thị tổng số sinh viên như được hiển thị bên dưới.

```
<label>Total Students:</label> @ViewBag.TotalStudents
```

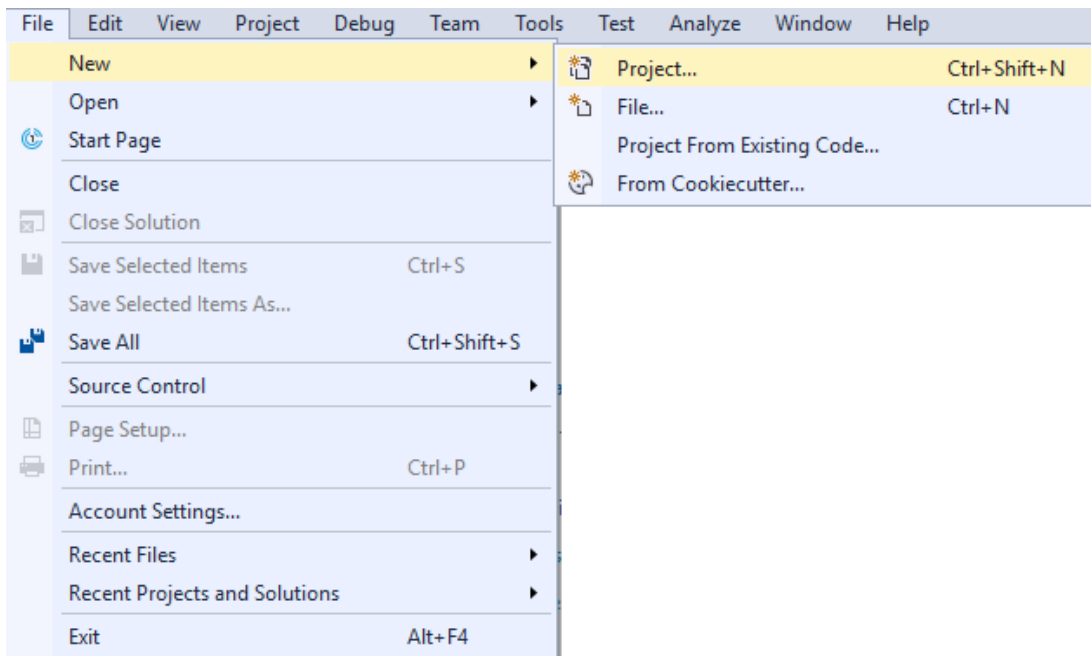
Kết quả:

```
Total Students: 5
```

ViewBag không yêu cầu ép kiểu trong khi lấy các giá trị.

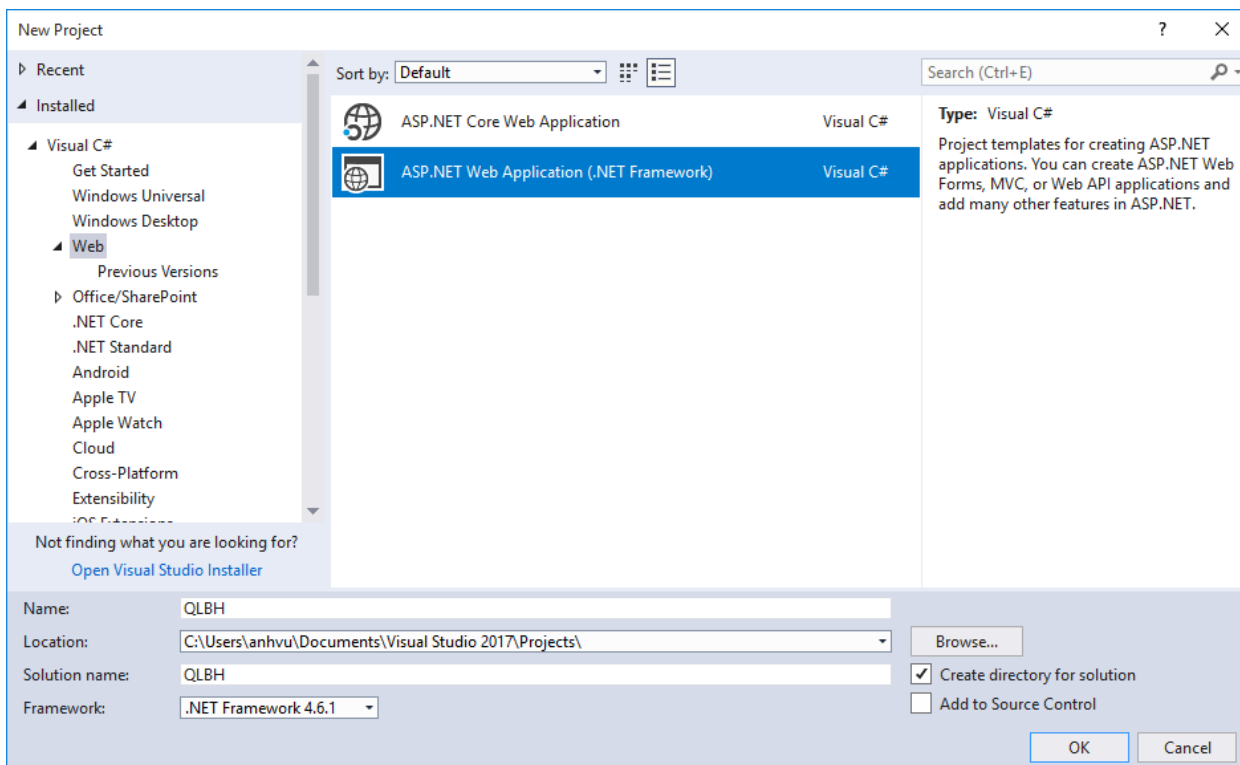
4/ Tạo project ASP.NET MVC:

- Mở Visual Studio ⇒ vào File ⇒ New ⇒ Project

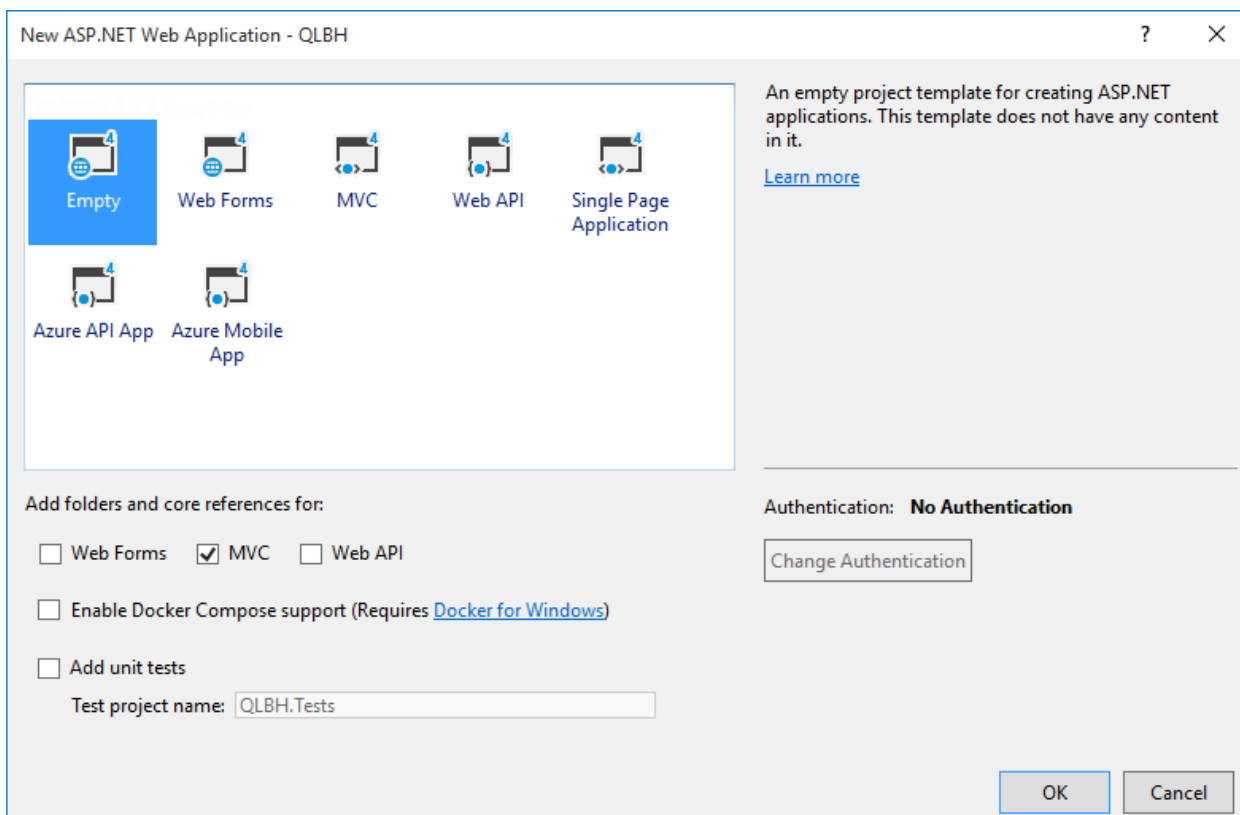


- Trong hộp thoại New Project, chọn Visual C# ⇒ Web ⇒ ASP.NET Web Application (.NET Framework) ⇒ Nhập tên project trong mục Name là QL BH ⇒ nhấn OK.

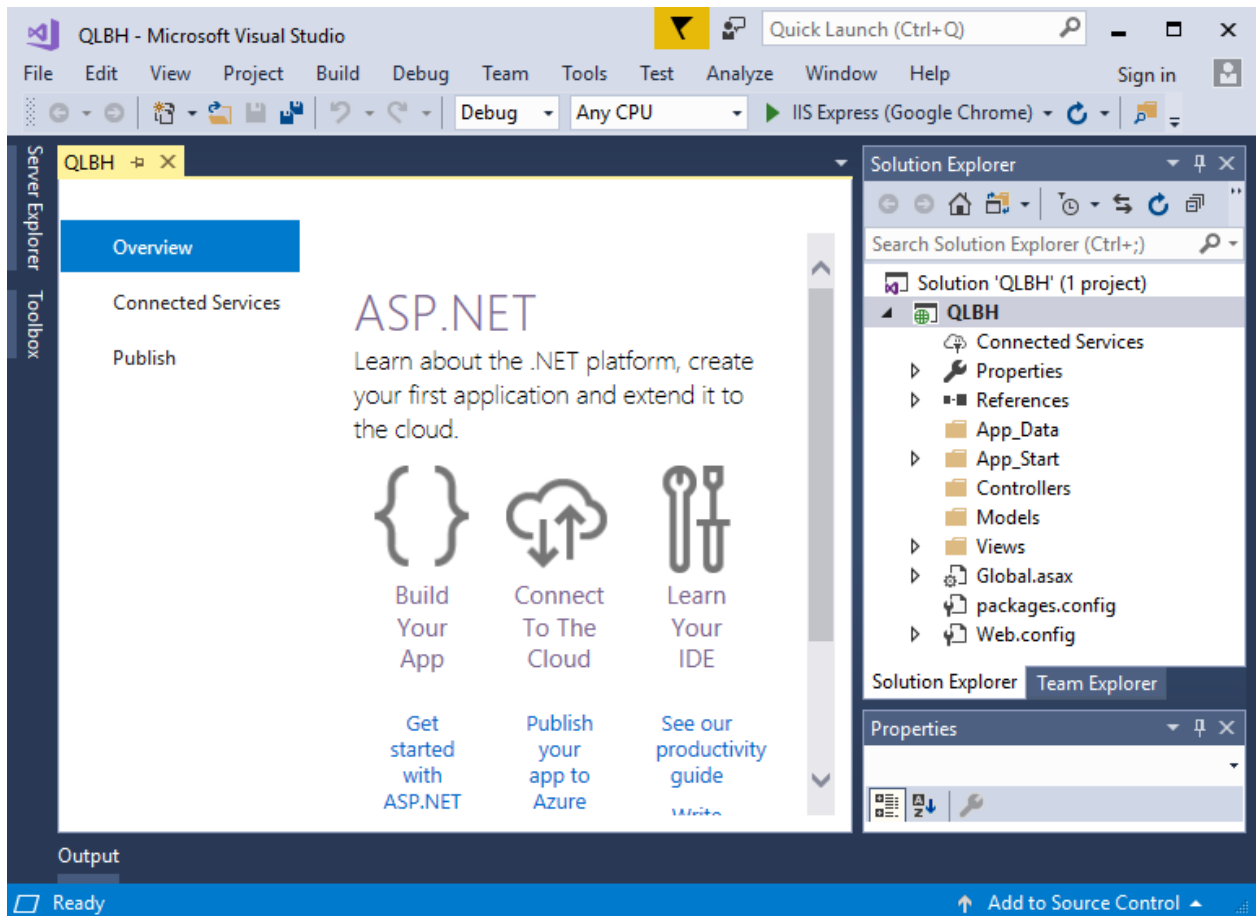
Bài 1, 2: ASP.NET MVC, HTML VÀ TRUY XUẤT CSDL BẰNG LINQ



- Trong hộp thoại New ASP.NET Web Application, chọn Empty ⇒ đánh dấu mục MVC ⇒ OK.

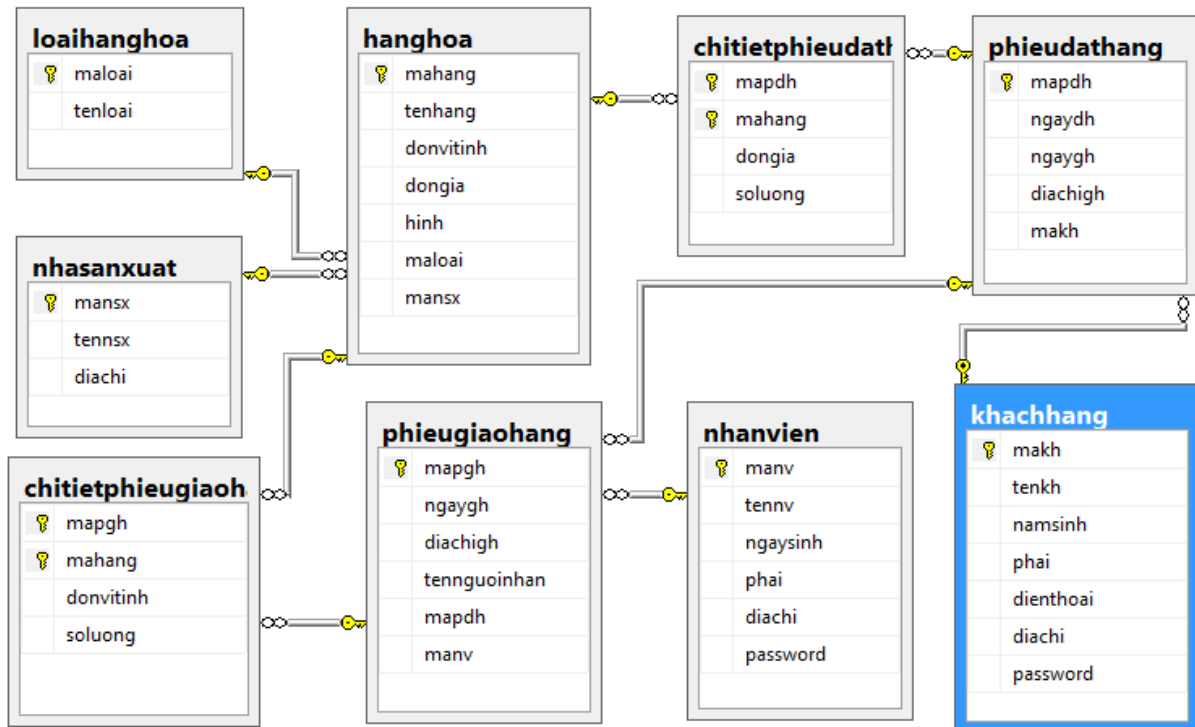


- Project đã được tạo:



5/ CSDL dùng cho các bài thực hành:

Bài 1, 2: ASP.NET MVC, HTML VÀ TRUY XUẤT CSDL BẰNG LINQ



Bảng hanghoa

ANHUU.QLBH - dbo.hanghoa X			
	Column Name	Data Type	Allow Nulls
	mahang	nvarchar(50)	<input type="checkbox"/>
	tenhang	nvarchar(50)	<input checked="" type="checkbox"/>
	donvitinh	nvarchar(50)	<input checked="" type="checkbox"/>
	dongia	float	<input checked="" type="checkbox"/>
	hinh	nvarchar(50)	<input checked="" type="checkbox"/>
	maloai	nvarchar(50)	<input checked="" type="checkbox"/>
	mansx	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng loaihanghoa

ANHUU.QLBH - dbo.loaihanghoa X			
	Column Name	Data Type	Allow Nulls
	maloai	nvarchar(50)	<input type="checkbox"/>
	tenloai	nvarchar(50)	<input checked="" type="checkbox"/>

Bảng nhasanxuat

ANHUU.QLBH - dbo.nhasanxuat X			
	Column Name	Data Type	Allow Nulls
	mansx	nvarchar(50)	<input type="checkbox"/>
	tennsx	nvarchar(50)	<input checked="" type="checkbox"/>
	diachi	nvarchar(50)	<input checked="" type="checkbox"/>

Bảng khachhang

ANHUU.QLBH - dbo.khachhang X			
	Column Name	Data Type	Allow Nulls
🔑	makh	nvarchar(50)	<input type="checkbox"/>
	tenkh	nvarchar(50)	<input checked="" type="checkbox"/>
	namsinh	int	<input checked="" type="checkbox"/>
	phai	bit	<input checked="" type="checkbox"/>
	dienthoai	nvarchar(50)	<input checked="" type="checkbox"/>
	diachi	nvarchar(50)	<input checked="" type="checkbox"/>
	password	nvarchar(50)	<input checked="" type="checkbox"/>

Bảng nhanvien

ANHUU.QLBH - dbo.nhanvien X			
	Column Name	Data Type	Allow Nulls
🔑	manv	nvarchar(50)	<input type="checkbox"/>
	tennv	nvarchar(50)	<input checked="" type="checkbox"/>
	ngaysinh	datetime	<input checked="" type="checkbox"/>
	phai	bit	<input checked="" type="checkbox"/>
	diachi	nvarchar(50)	<input checked="" type="checkbox"/>
	password	nvarchar(50)	<input checked="" type="checkbox"/>

Bảng phieudathang

ANHUU.QLBH - dbo.phieudathang X			
	Column Name	Data Type	Allow Nulls
🔑	mapdh	nvarchar(50)	<input type="checkbox"/>
	ngaydh	datetime	<input checked="" type="checkbox"/>
	ngaygh	datetime	<input checked="" type="checkbox"/>
	diachigh	nvarchar(50)	<input checked="" type="checkbox"/>
	makh	nvarchar(50)	<input checked="" type="checkbox"/>

Bảng chitietphieudathang

ANHUU.QLBH - dbo....itietphieudathang X			
	Column Name	Data Type	Allow Nulls
🔑	mapdh	nvarchar(50)	<input type="checkbox"/>
🔑	mahang	nvarchar(50)	<input type="checkbox"/>
	dongia	float	<input checked="" type="checkbox"/>
	soluong	float	<input checked="" type="checkbox"/>

Bảng phieugiaohang

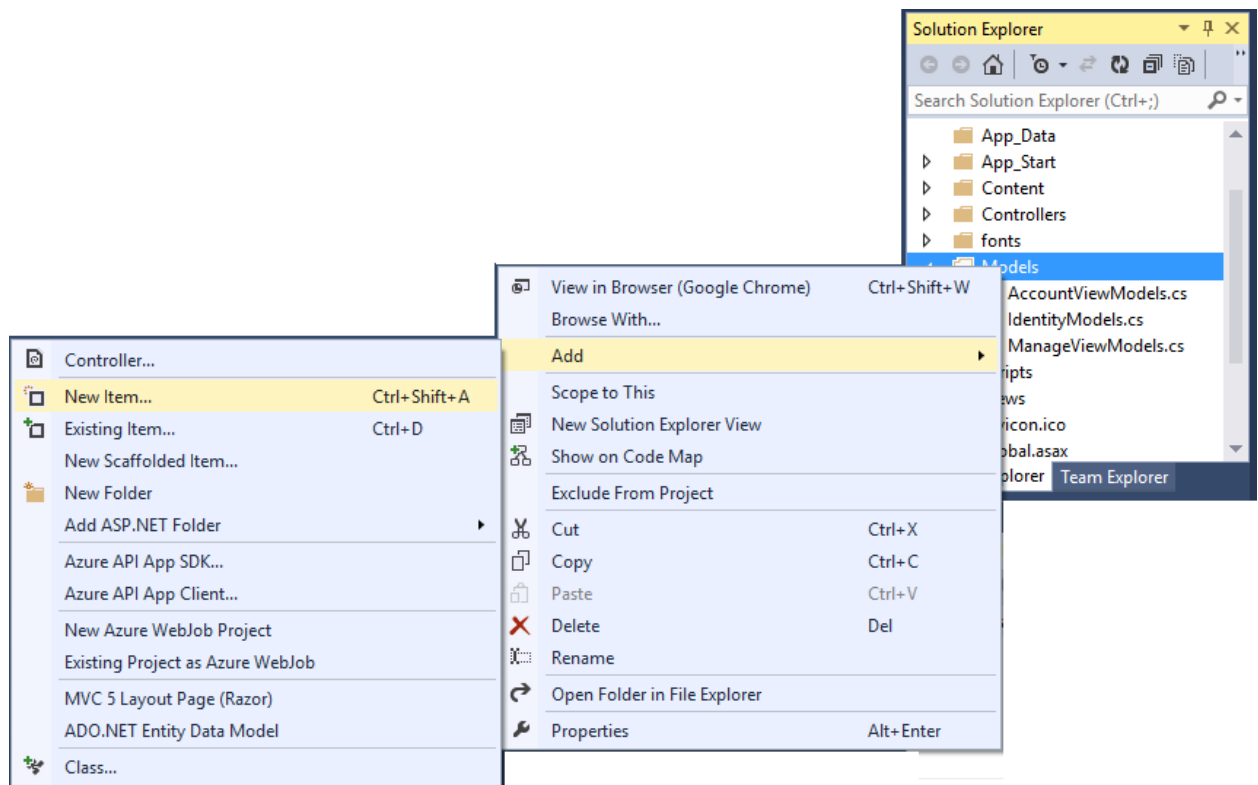
ANHUU.QLBH - dbo.phieugiaohang X			
	Column Name	Data Type	Allow Nulls
🔑	mapgh	nvarchar(50)	<input type="checkbox"/>
	ngaygh	datetime	<input checked="" type="checkbox"/>
	diachigh	nvarchar(50)	<input checked="" type="checkbox"/>
	tennguoinhan	nvarchar(50)	<input checked="" type="checkbox"/>
	mapdh	nvarchar(50)	<input checked="" type="checkbox"/>
	manv	nvarchar(50)	<input checked="" type="checkbox"/>

Bảng chitietphieugiaohang

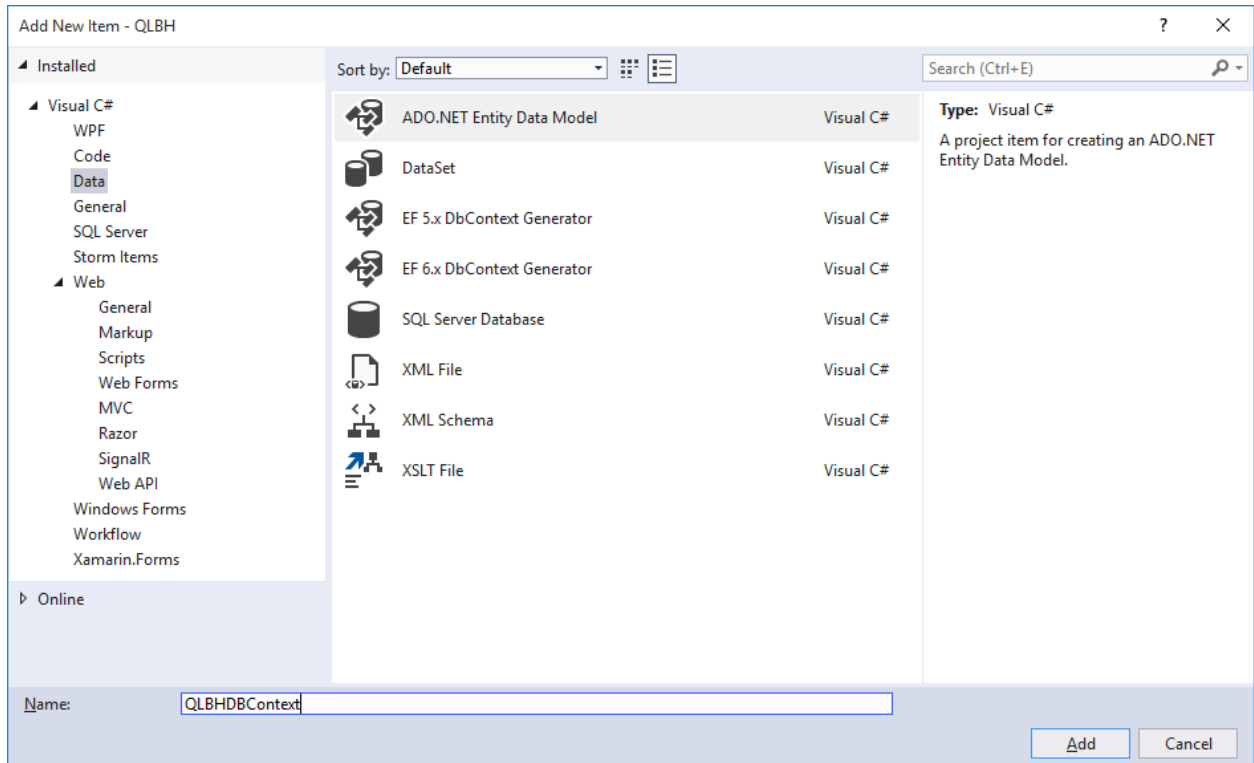
ANHUU.QLBH - dbo....itietphieugiaohang X			
	Column Name	Data Type	Allow Nulls
🔑	mapdh	nvarchar(50)	<input type="checkbox"/>
🔑	mahang	nvarchar(50)	<input type="checkbox"/>
	dongia	float	<input checked="" type="checkbox"/>
	soluong	float	<input checked="" type="checkbox"/>

6/ Sử dụng ADO.NET Entity Data Model

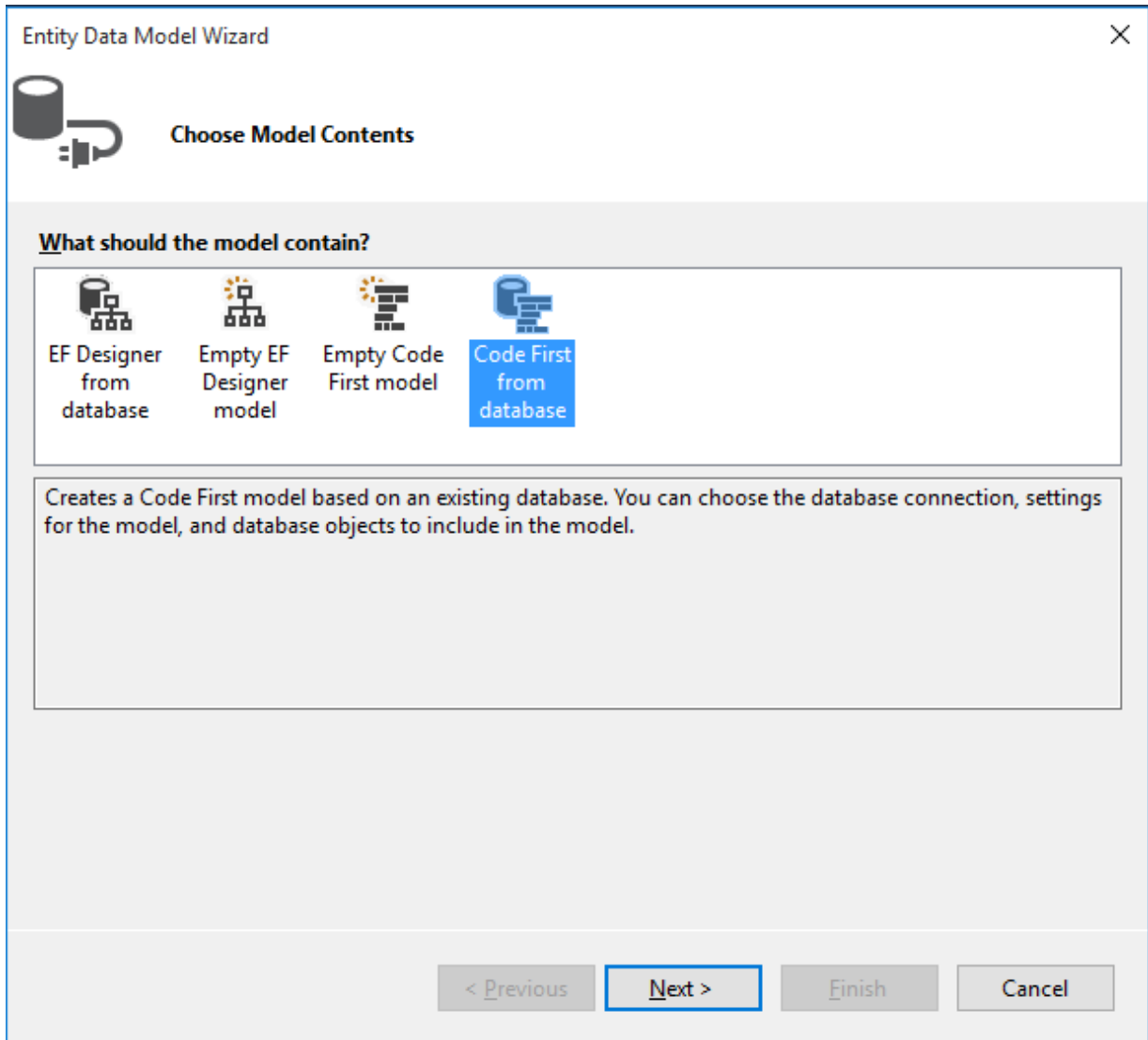
- Click phải vào thư mục Models ⇒ Add ⇒ New Item



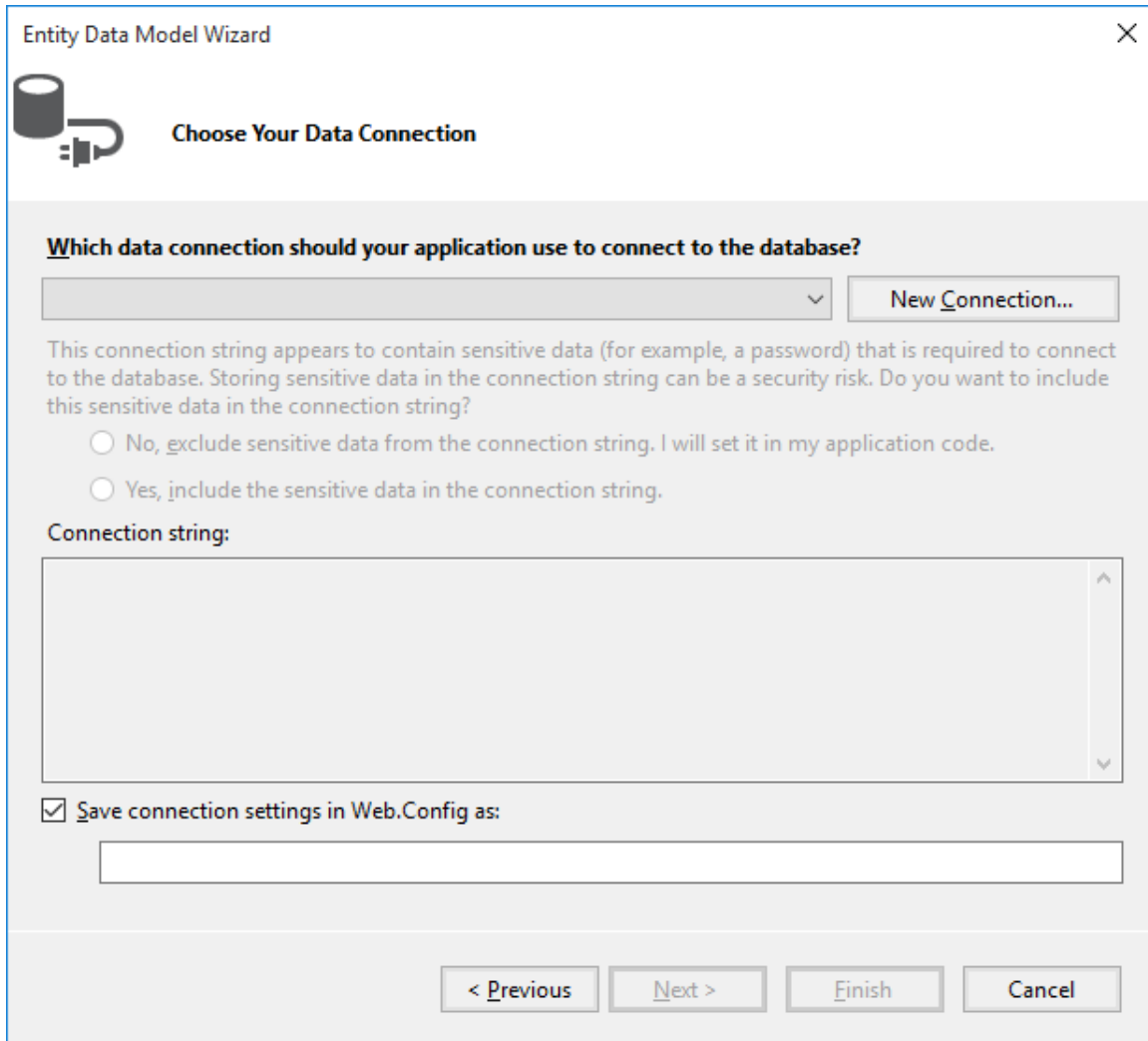
- Trong cửa sổ Add New Item, chọn Installed ⇒ Visual C# ⇒ Data ⇒ ADO.NET Entity Data Model ⇒ nhập trong mục Name là **QLBHDBContext** ⇒ nhấn Add.



- Trong cửa sổ Entity Data Model Wizard, chọn Code First from database ⇒ nhấn Next.



- Trong cửa sổ Choose Your Data Connection, chọn New Connection.



The image shows the 'Entity Data Model Wizard' dialog box, specifically the 'Choose Your Data Connection' step. It features a title bar with a close button (X). Below the title bar is a header area with a database icon and the text 'Choose Your Data Connection'. The main content area asks, 'Which data connection should your application use to connect to the database?'. It includes a dropdown menu, a 'New Connection...' button, and a warning message about sensitive data in the connection string. There are two radio buttons for choosing whether to include sensitive data. Below this is a 'Connection string:' label and a large text area. At the bottom, there is a checkbox labeled 'Save connection settings in Web.Config as:' followed by a text input field. The footer contains four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

☒ Save connection settings in Web.Config as:

< Previous Next > Finish Cancel

- Trong cửa sổ Connection Properties, chúng ta nhập thông tin database đã tạo ở phần trên vào
⇒ nhấn OK.

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
localhost Refresh

Log on to the server

☒ Use Windows Authentication
☐ Use SQL Server Authentication

User name:
Password:
☐ Save my password

Connect to a database

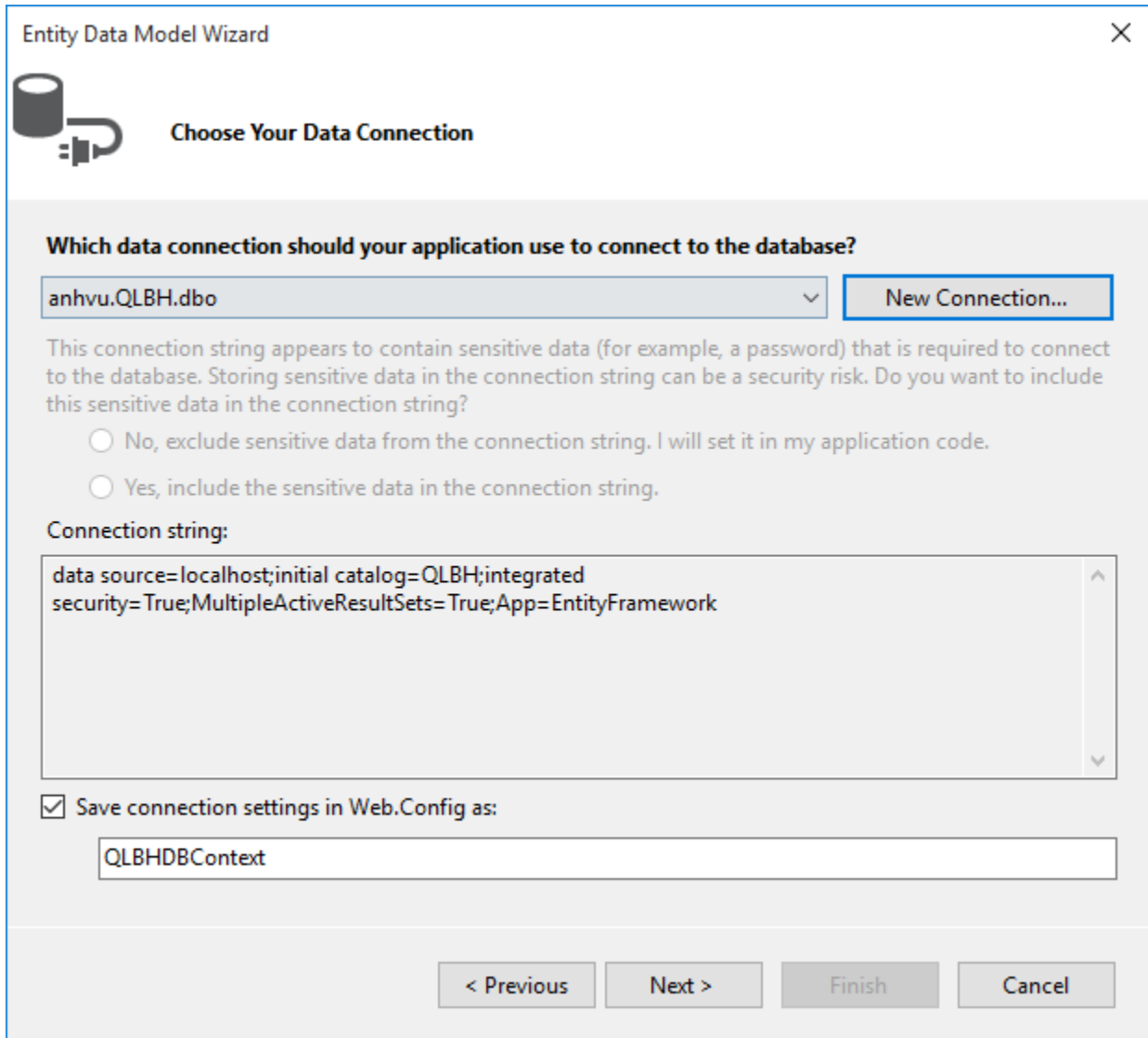
☒ Select or enter a database name:
QLBH

☐ Attach a database file:
 Browse...
Logical name:

Advanced...

Test Connection OK Cancel

- Trong cửa sổ Choose Your Data Connection, nhấn Next.



The image shows the 'Entity Data Model Wizard' window, specifically the 'Choose Your Data Connection' step. The window has a title bar with a close button. Below the title bar is a header area with a database icon and the text 'Choose Your Data Connection'. The main content area asks 'Which data connection should your application use to connect to the database?'. There is a dropdown menu showing 'anhvu.QLBH.dbo' and a 'New Connection...' button. Below this, a message states: 'This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?'. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' and 'Yes, include the sensitive data in the connection string.'. Below the radio buttons is a text box labeled 'Connection string:' containing the string: 'data source=localhost;initial catalog=QLBH;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework'. At the bottom, there is a checkbox labeled 'Save connection settings in Web.Config as:' which is checked, and a text box containing 'QLBHDBContext'. At the very bottom are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

anhvu.QLBH.dbo

New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

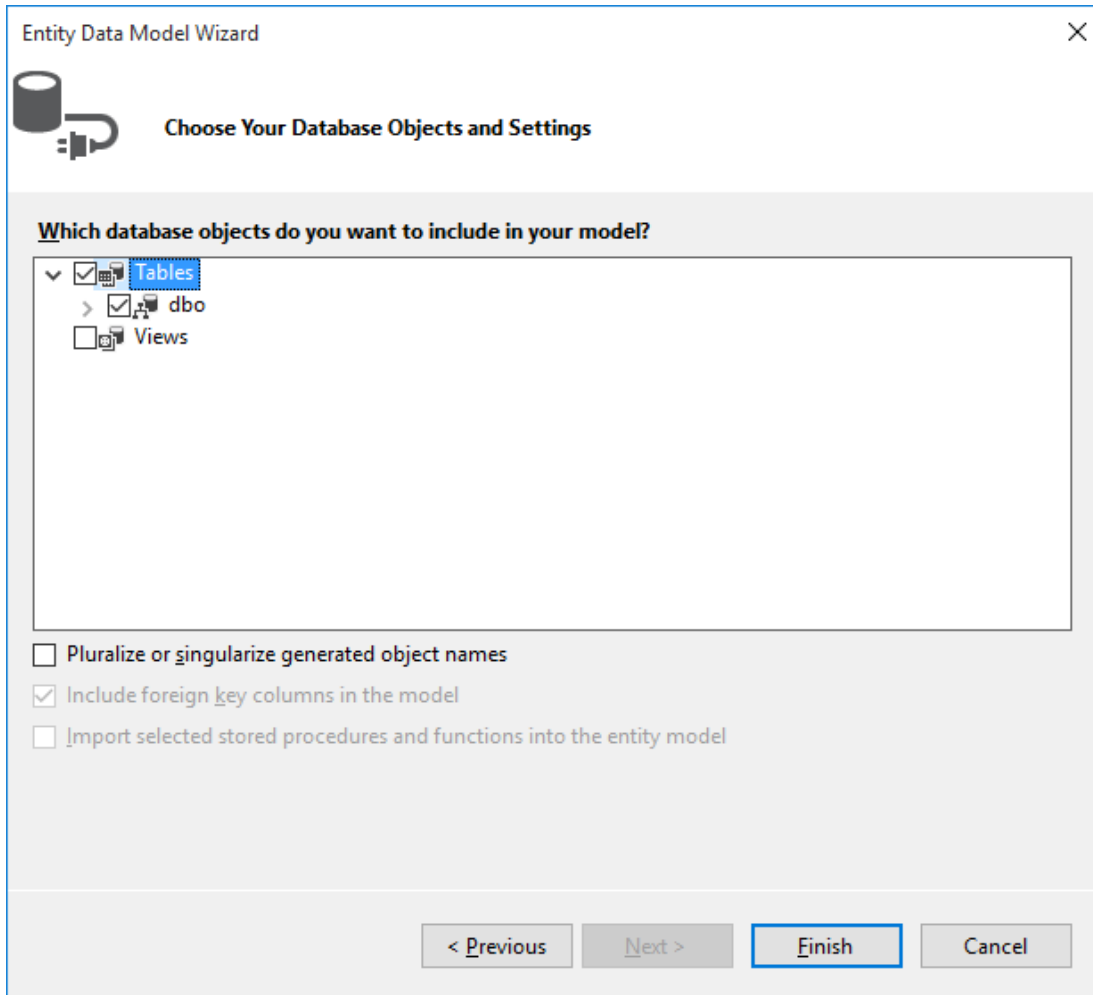
data source=localhost;initial catalog=QLBH;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework

☒ Save connection settings in Web.Config as:

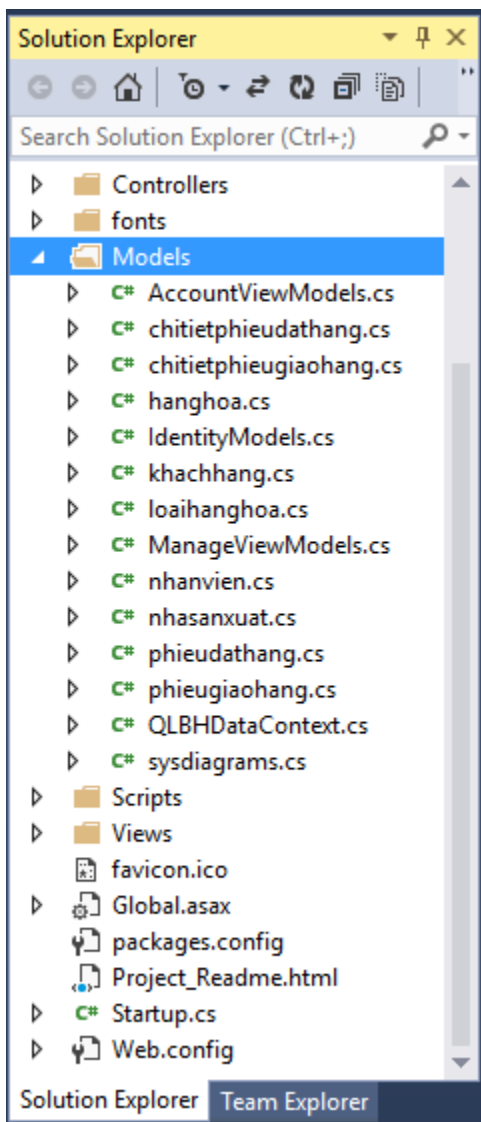
QLBHDBContext

< Previous Next > Finish Cancel

- Trong cửa sổ Choose Your Database Objects and Settings, nhấn đánh dấu Tables ⇒ Finish.



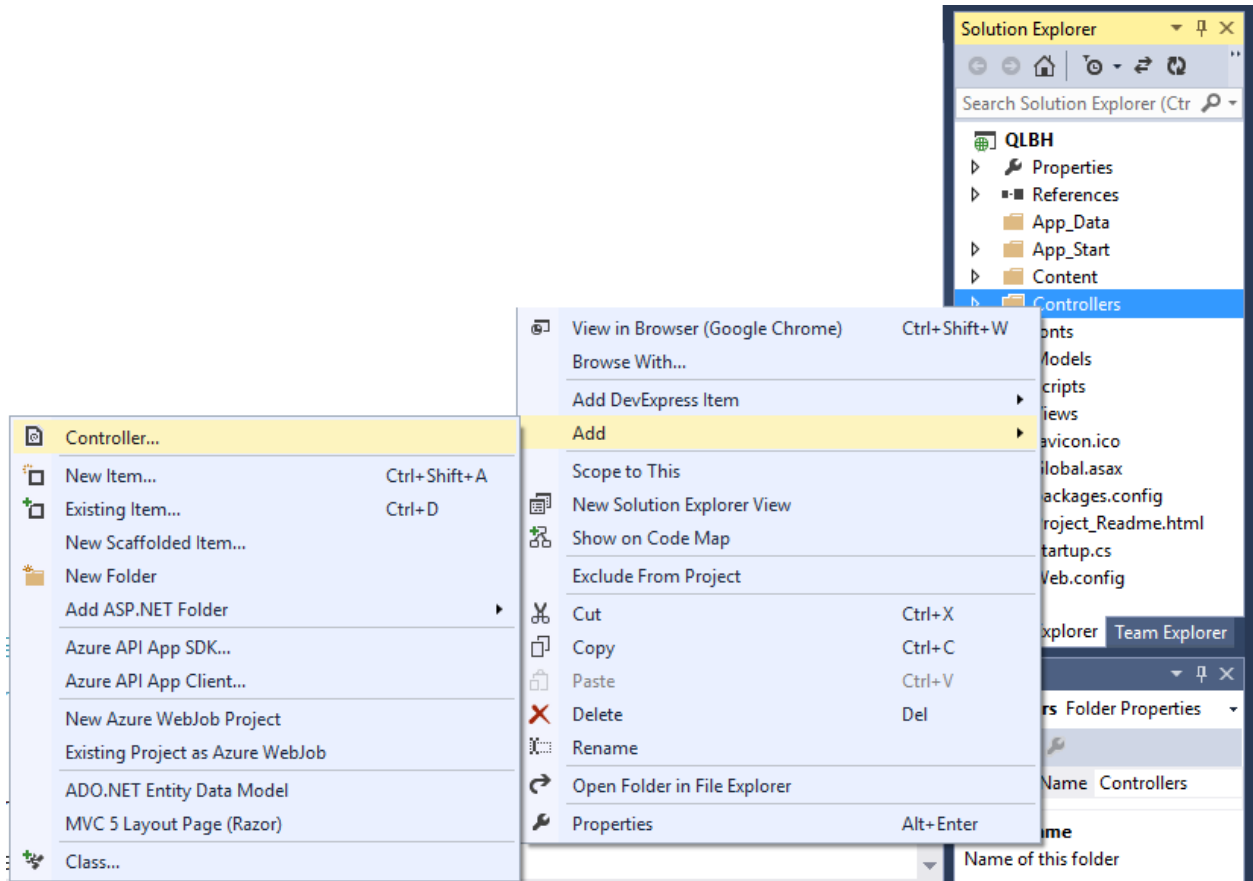
- Ta thấy các lớp Model sẽ được tạo ra trong thư mục Models.



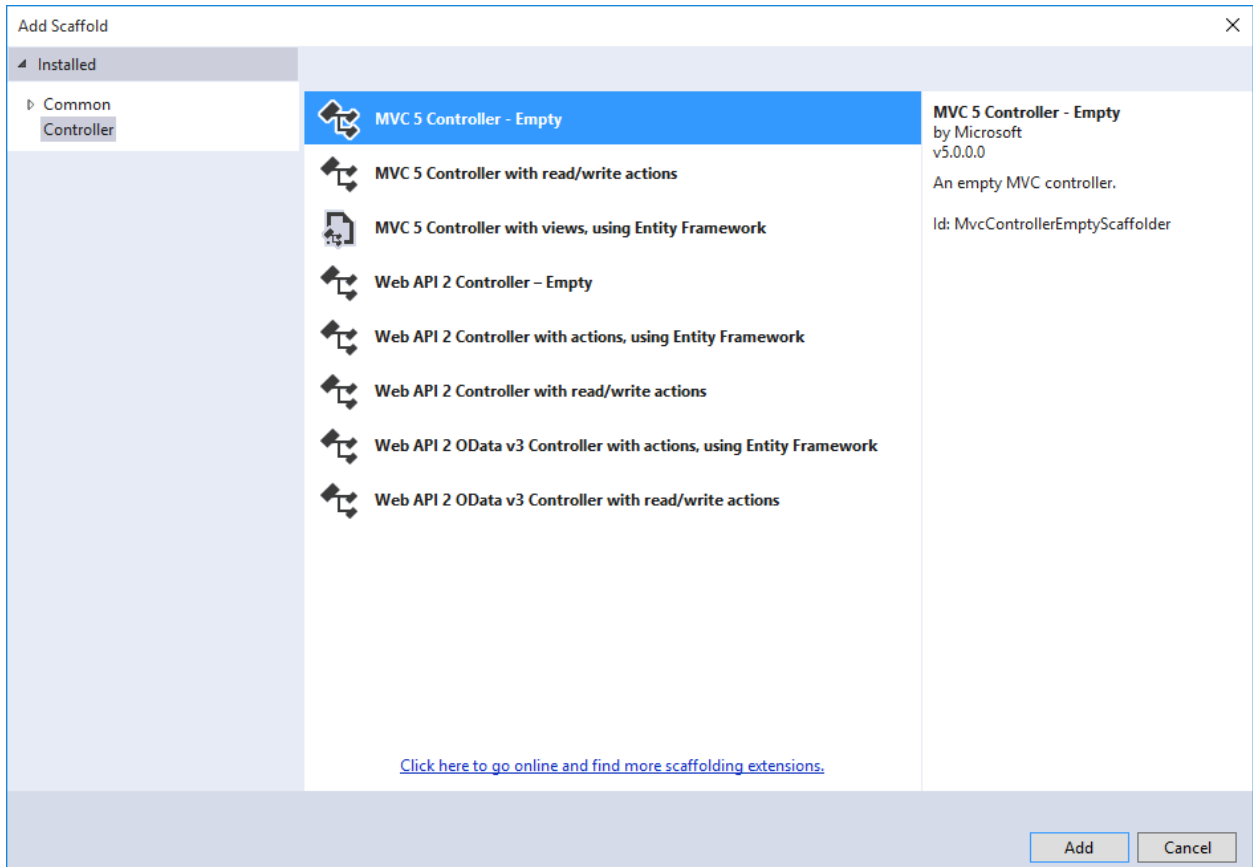
III. NỘI DUNG THỰC HÀNH BÀI 1:

1/ Chương trình mẫu:

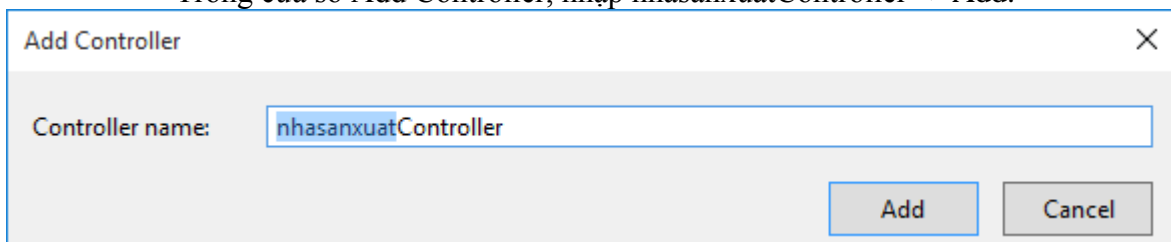
- a) Tạo project ASP.NET như trong phần hướng dẫn lý thuyết với tên project là **QLBH**.
- b) Thêm ADO.NET Entity Data Model như trong phần hướng dẫn lý thuyết với connection string là **QLBHDBContext**.
- c) Thực hiện chức năng hiển thị nhà sản xuất :
 - ❖ **Thêm Controller cho model nhasanxuat:**
 - Click phải vào thư mục Controller trong cửa sổ Solution Explorer ⇒ Add ⇒ Controller.



- Trong cửa sổ Add Scaffold, chọn Controller ⇒ MVC 5 Controller – Empty ⇒ Add.



- Trong cửa sổ Add Controller, nhập nhasanxuatController ⇒ Add.

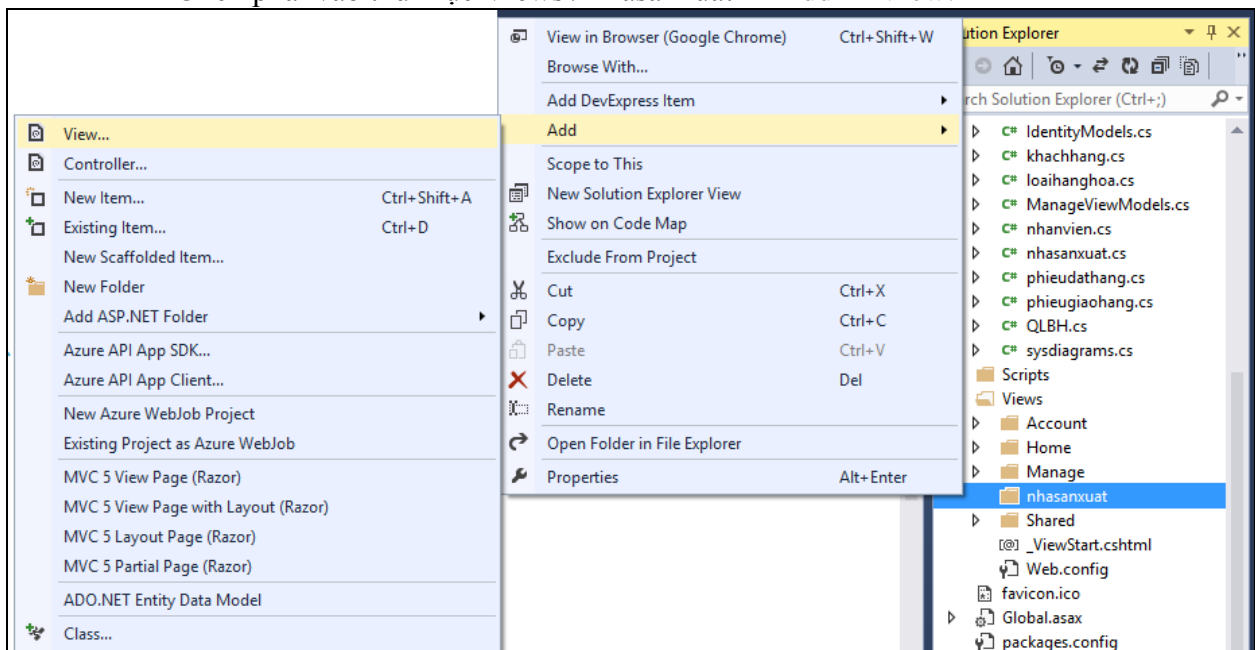


- Một controller mới sẽ được thêm vào, viết thêm code :

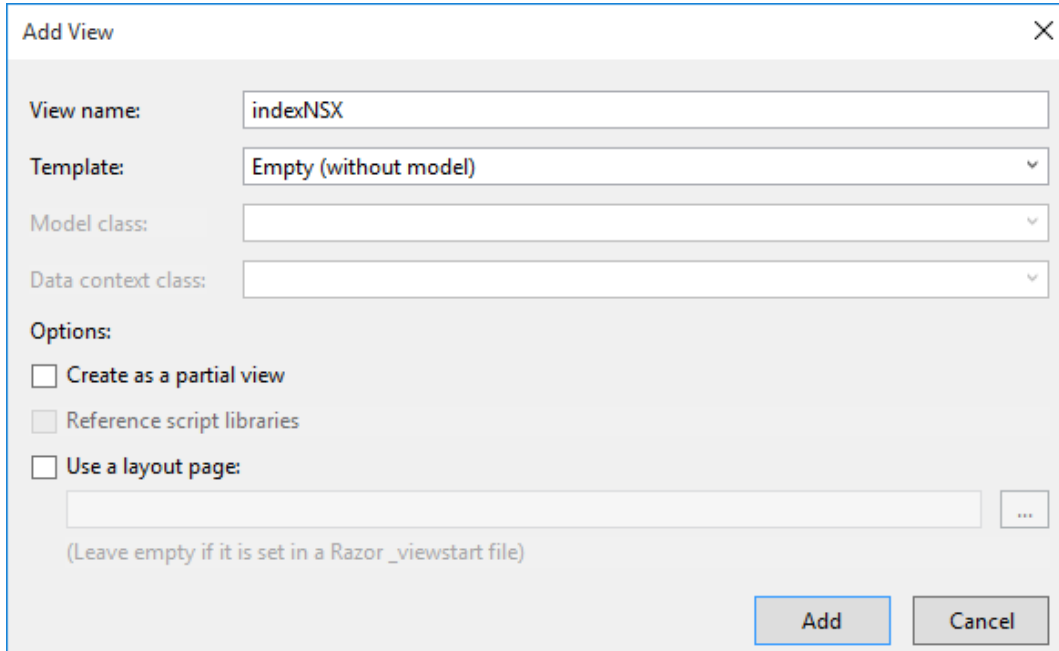
```
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Net; //Thêm vào
using QLBH.Models;
using System.Data;
using System.Data.Entity;
namespace QLBH.Controllers
{
    0 references
    public class nhasanxuatController : Controller
    {
        QLBHDBContext db = new QLBHDBContext();
        // GET: nhasanxuat
        0 references
        public ActionResult indexNSX()
        {
            ViewBag.nsx = db.nhasanxuat;
            return View();
        }
    }
}
```

❖ **Thêm View cho nhasanxuat:**

- Click phải vào thư mục Views / nhasanxuat ⇒ Add ⇒ View.



- Trong cửa sổ Add View, để thêm View **indexNSX** cho nhasanxuat, ta nhập thông như trong hình ⇒ Add.



- Kết quả ta được :

```
1
2  @{
3      Layout = null;
4  }
5
6  <!DOCTYPE html>
7
8  <html>
9  <head>
10     <meta name="viewport" content="width=device-width" />
11     <title>Index</title>
12 </head>
13 <body>
14     <div>
15     </div>
16 </body>
17 </html>
18
```

- Trong trang indexNSX.cshtml, SV hãy nhập lại :

```
@{
    Layout = null;
}

<h2>Thông tin các nhà sản xuất</h2>

<p>
```

```
<a href="@Url.Action("themNSX")">Thêm mới</a>
</p>

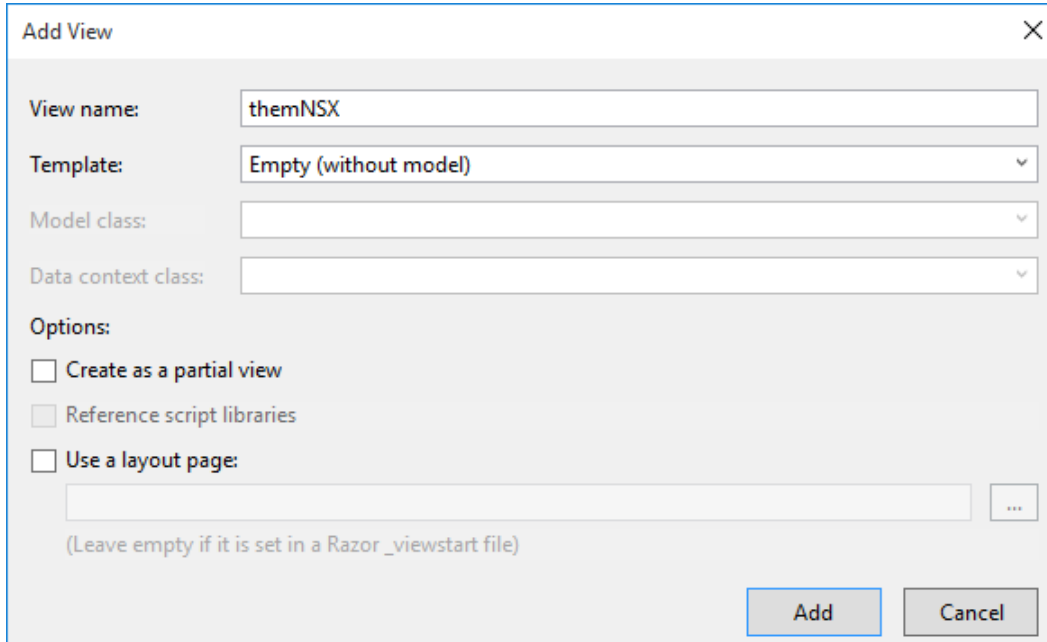
<table border="1" cellpadding="5" cellspacing="0">
    <tr>
        <th>Tên nhà sản xuất</th>
        <th>Địa chỉ</th>
    </tr>
    @{
        foreach (var item in ViewBag.nsx)
        {
            <tr>
                <td>@item.tennsx</td>
                <td>@item.diachi</td>
            </tr>
        }
    }
</table>
```

- Kết quả thực thi trang indexNSX.cshtml :

Thông tin các nhà sản xuất	
Thêm mới	
Tên nhà sản xuất	Địa chỉ
Công ty ASUS Việt nam	123 Trần Hưng Đạo
Công ty Intel Việt Nam	222 Cách Mạng Tháng 8

d) SV thực hiện chức năng thêm mới nhà sản xuất.

❖ Thêm 1 view là **themNSX** trong thư mục **nhasanxuat** như sau :



- Nhấn Add.
- Viết code cho themNSX.cshtml :

```
@{
    Layout = null;
}

<form action="themNSX" method="post">
    <div class="form-horizontal">
        <h2>Thêm nhà sản xuất</h2>

        <div class="form-group">
            @*name trùng với tên field được binding*@
            <b>Mã nhà sản xuất</b><br />
            <input name="mansx" type="text" value="" maxlength="50" /><br
        /><br />
        </div>

        <div class="form-group">
            <b>Tên nhà sản xuất</b><br />
            <input name="tennsx" type="text" value="" maxlength="50" /><br
        /><br />
        </div>

        <div class="form-group">
            <b>Địa chỉ</b><br />
            <input name="diachi" type="text" value="" maxlength="50" /><br
        /><br />
        </div>
    </div>
</form>
```

```
<div class="form-group">
    <input type="submit" value="Save" /><br /><br />
</div>

<div>
    <a href="@Url.Action("indexNSX")">Quay lại</a>
</div>
</div>
</form>
```

- Kết quả thực thi :



❖ Trong **nhasanxuatController.cs**, SV hãy viết code :

```
[HttpGet]
public ActionResult themNSX()
{
    return View();
}
[HttpPost]
public ActionResult themNSX(nhasanxuat n)
{
    db.nhasanxuat.Add(n);
    db.SaveChanges();
    return RedirectToAction("indexNSX");
}
```

❖ SV hãy thực thi để hiển thị và thêm mới nhà sản xuất.

2/ Bài tập tổng hợp:

- SV hãy thực hiện chức năng hiển thị và thêm mới cho chức năng quản lý loại hàng hóa.
- SV hãy thực hiện chức năng hiển thị và thêm mới cho chức năng quản lý nhân viên.

3/ Bài tập làm thêm:

- a) SV hãy thực hiện chức năng hiển thị và thêm mới cho chức năng quản lý khách hàng.
- b) SV hãy thực hiện chức năng hiển thị và thêm mới cho chức năng quản lý hàng hóa.

IV. NỘI DUNG THỰC HÀNH BÀI 2:**1/ Chương trình mẫu:**

SV hãy mở project của Bài 1 và làm tiếp.

a) Thực hiện chức năng sửa nhà sản xuất

❖ Trong indexNSX.cshtml, ta thêm cột “Tác vụ” như sau:

```
@{
    Layout = null;
}

<h2>Thông tin các nhà sản xuất</h2>

<p>
    <a href="@Url.Action("themNSX")">Thêm mới</a>
</p>

<table border="1" cellpadding="5" cellspacing="0">
    <tr>
        <th>Tên nhà sản xuất</th>
        <th>Địa chỉ</th>
        <th>Tác vụ</th>
    </tr>
    @{
        foreach (var item in ViewBag.nsx)
        {
            <tr>
                <td>@item.tennsx</td>
                <td>@item.diachi</td>
                <td>
                    <a href="@Url.Action("suaNSX","nhasanxuat",new { id =
item.mansx })">Sửa</a> |
                    <a href="@Url.Action("xoaNSX","nhasanxuat",new { id =
item.mansx })">Xóa</a>
                </td>
            </tr>
        }
    }
</table>
```

Kết quả hiển thị :

Thông tin các nhà sản xuất

[Thêm mới](#)

Tên nhà sản xuất	Địa chỉ	Tác vụ
Công ty ASUS Việt nam	123 Trần Hưng Đạo	Sửa Xóa
Công ty Intel Việt Nam	222 Cách Mạng Tháng 8	Sửa Xóa

❖ Thêm view **suaNSX** vào thư mục nhasanxuat và viết code sau :

```
@using QLBH.Models;
@{
    Layout = null;
}

<form action="suaNSX" method="post">
    <div class="form-horizontal">
        <h2>Sửa thông tin nhà sản xuất</h2>
        @{
            nhasanxuat n = ViewBag.nsx;
        }
        <div class="form-group">
            @*name trùng với tên field được binding*@
            <input name="mansx" type="hidden" value="@n.mansx" maxlength="50"
        /><br />
        </div>

        <div class="form-group">
            <b>Tên nhà sản xuất</b><br />
            <input name="tennsx" type="text" value="@n.tennsx" maxlength="50"
        /><br />
        </div>
        <br />
        <div class="form-group">
            <b>Địa chỉ</b><br />
            <input name="diachi" type="text" value="@n.diachi" maxlength="50"
        /><br />
        </div>
        <br />
        <div class="form-group">
            <input type="submit" value="Lưu" />
        </div>
        <br />
        <div>
            <a href="@Url.Action("indexNSX")">Quay lại</a>
        </div>
    </div>
</form>
```

❖ Trong **nhasanxuatController.cs**, SV viết code cho action **suaNSX** như sau :

```
[HttpGet]
public ActionResult suaNSX(string id)
{
    if (id == null)
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    nhasanxuat nsx = db.nhasanxuat.Find(id);
    if (nsx == null)
        return HttpNotFound();
    ViewBag.nsx = nsx;
    return View(nsx);
}

[HttpPost]
public ActionResult suaNSX()
{
    string ma = Request["mansx"].ToString();
    nhasanxuat nsx = db.nhasanxuat.Find(ma);
    if (nsx == null)
        return HttpNotFound();
    nsx.tennsx = Request["tennsx"].ToString();
    nsx.diachi = Request["diachi"].ToString();
    db.Entry(nsx).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("indexNSX");
}
```

❖ SV hãy thực thi và xem kết quả.

b) Thực hiện chức năng xóa nhà sản xuất

❖ Thêm view **xoaNSX** vào thư mục **nhasanxuat** và viết code sau :

```
@using QBH.Models
@{
    Layout = null;
}
<h2>Xóa thông tin nhà sản xuất</h2>
<h3>Bạn muốn xóa nhà sản xuất này ?</h3>

<form method="post">
    <div class="form-group">
        <table border="1" cellpadding="5" cellspacing="0">
            <tr>
                <th>Tên nhà sản xuất</th>
                <th>Địa chỉ</th>
            </tr>
            <tr>
                <td><input type="text" value="@ViewBag.nsx.tennsx" />
                <td><input type="text" value="@ViewBag.nsx.diachi" />
            </tr>
        </table>
    </div>
    <div>
        <input type="button" value="Xóa" />
        <input type="button" value="Hủy" />
    </div>
</form>
```

```
        nhasanxuat n = ViewBag.nsx;  
    }  
    <td>  
        <input type="hidden" name="mansx" value="@n.mansx" />  
        <label for="tennsx">@n.tennsx</label>  
    </td>  
    <td>  
        <label for="diachi">@n.diachi</label>  
    </td>  
</tr>  
</table>  
</div>  
<br />  
<div class="form-group">  
    <input type="submit" value="Xóa" />  
</div>  
<br />  
<div>  
    <a href="@Url.Action("indexNSX")">Quay lại</a>  
</div>  
</form>
```

❖ Trong **nhasanxuatController.cs**, SV viết code cho action **xoaNSX** như sau :

```
public ActionResult xoaNSX(string id)  
{  
    if (id == null)  
    {  
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  
    }  
    nhasanxuat n = db.nhasanxuat.Find(id);  
    if (n == null)  
    {  
        return HttpNotFound();  
    }  
    ViewBag.nsx = n;  
    return View(n);  
}  
  
[HttpPost, ActionName("xoaNSX")]  
public ActionResult xoaNSX_Post(string id)  
{  
    nhasanxuat n = db.nhasanxuat.Find(id);  
    db.nhasanxuat.Remove(n);  
    db.SaveChanges();  
    return RedirectToAction("indexNSX");  
}
```

2/ Bài tập tổng hợp:

- SV hãy thực hiện chức năng sửa và xóa cho chức năng quản lý loại hàng hóa.
- SV hãy thực hiện chức năng sửa và xóa cho chức năng quản lý nhân viên.

3/ Bài tập làm thêm:

- a) SV hãy thực hiện chức năng sửa và xóa cho chức năng quản lý khách hàng.
- b) SV hãy thực hiện chức năng sửa và xóa cho chức năng quản lý hàng hóa.

I. MỤC TIÊU:

Sau khi thực hành bài này, SV có thể :

- Hiểu và vận dụng Bootstrap, HtmlHelper.
- Đối tượng Model, tham số của phương thức Action.
- Thẻ Check, Radio, Select. Layout view.
- Thao tác trên 1 Table trong CSDL.
- Thực hiện các phép toán hiển thị danh sách, thêm, xóa, sửa.

II. TÓM TẮT LÝ THUYẾT:

1/ Bootstrap:

a) **Thêm Bootstrap vào project:** có các cách để thêm Bootstrap vào project:

- ❖ Cách 1: dựa vào CDN (mạng phân phối nội dung)

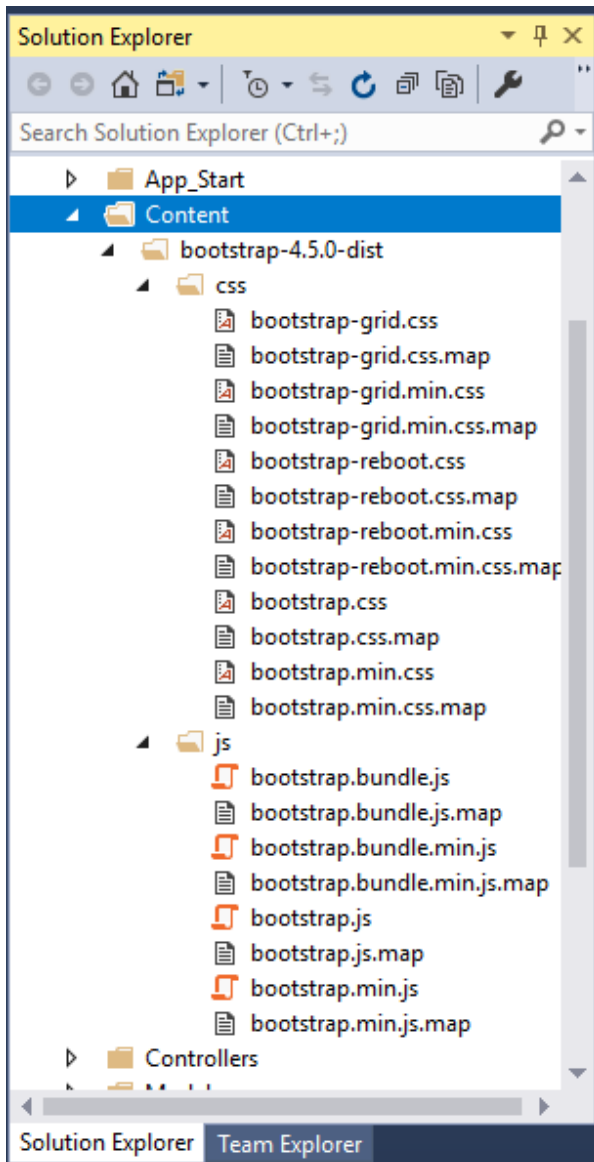
Ví dụ:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
</head>
<body>

</body>
</html>
```

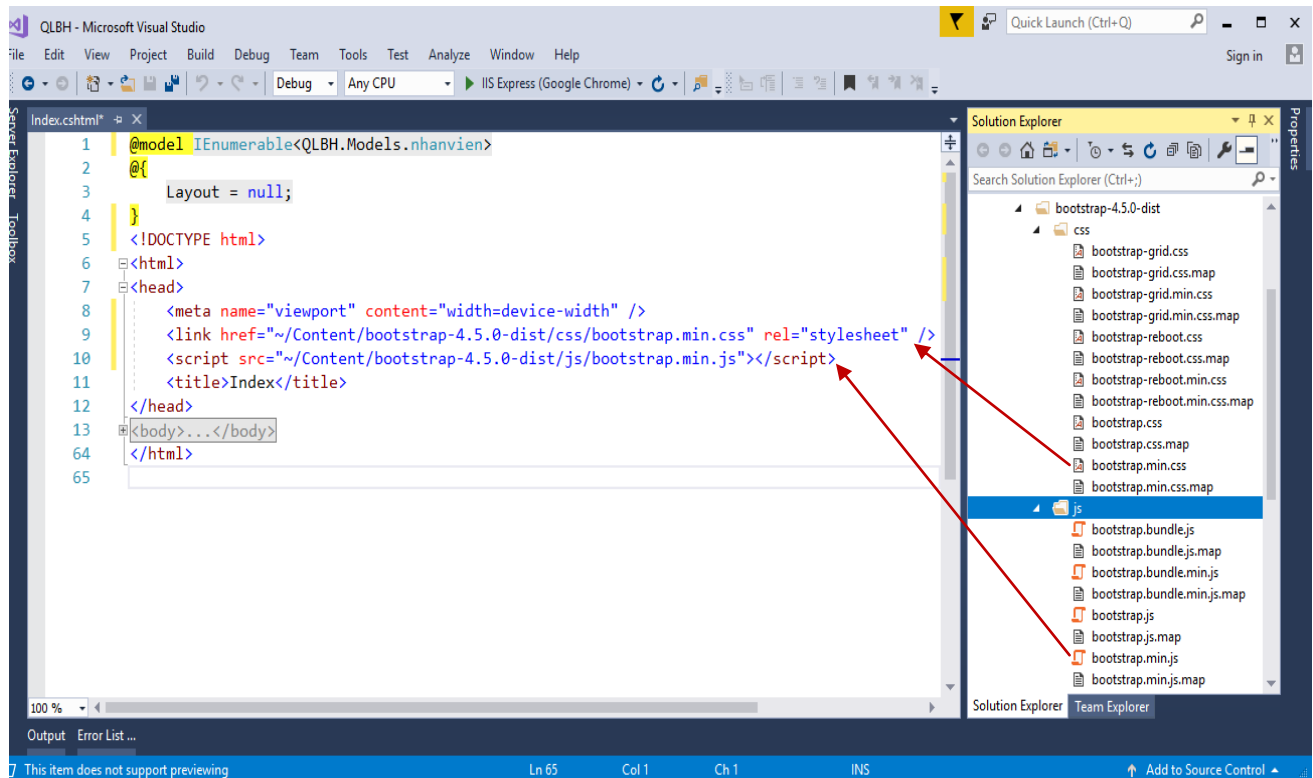
- ❖ Cách 2: tải Bootstrap trên mạng và thêm vào project
- Tải Bootstrap trên trang web <https://getbootstrap.com>. Sau khi giải nén file đã tải, chúng ta sao chép vào project :

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ



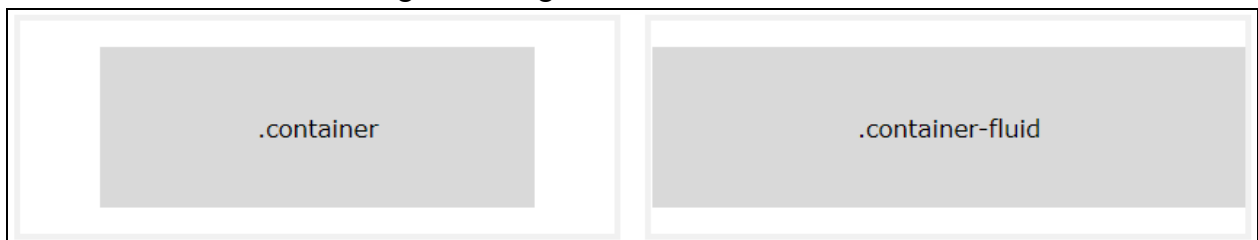
- Kéo 2 file này vào trang cần dùng :

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ



b) **Containers** : để bao bọc nội dung trang web.

- Lớp .container cung cấp một thành phần chứa có chiều rộng cố định
- Lớp .container-fluid cung cấp một thành phần chứa có chiều rộng đầy đủ, trải rộng toàn bộ chiều rộng của khung nhìn.



c) **Hệ thống lưới**: hệ thống lưới của Bootstrap cho phép tối đa 12 cột trên trang.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ

Hệ thống lưới của Bootstrap có 5 lớp. Các lớp ở trên có thể được kết hợp để trang web năng động và linh hoạt hơn.

.col- (extra small devices - screen width less than 576px)

.col-sm- (small devices - screen width equal to or greater than 576px)

.col-md- (medium devices - screen width equal to or greater than 768px)

.col-lg- (large devices - screen width equal to or greater than 992px)

.col-xl- (xlarge devices - screen width equal to or greater than 1200px)

Cấu trúc cơ bản của lưới Bootstrap: <div class="row"> tạo dòng, còn <div class="col-*-*"> tạo cột.

VD:

```
<div class="row">
  <div class="col-sm-4" style="background-color:lavender;">.col-sm-4</div>
  <div class="col-sm-8" style="background-color:lavenderblush;">.col-sm-8</div>
</div>
```

Kết quả:

.col-sm-4	.col-sm-8
-----------	-----------

d) Bảng Bootstrap:

- Lớp **.table** thêm kiểu dáng cơ bản vào bảng.
- Lớp **.table-striped** thêm sọc ngang vào bảng.
- Lớp **.table-bordered** thêm đường viền trên tất cả các mặt của bảng và các ô.
- Lớp **.table-hover** thêm hiệu ứng di chuyển chuột (màu nền xám) trên các hàng của bảng.
- Lớp **.table-dark** thêm nền đen vào bảng.
- Các lớp ngữ cảnh để tô màu cho toàn bộ bảng (<table>), các hàng của bảng (<tr>) hoặc các ô của bảng (<td>): **.table-primary**, **.table-success**, **.table-danger**, **.table-info**, **.table-warning**, **.table-active**, **.table-secondary**, **.table-light**, **.table-dark**.

e) List Groups:

- Sử dụng thẻ với lớp **.list-group** để tạo List Group cơ bản và các phần tử của List Groups dùng thẻ với lớp **.list-group-item**.

```
<div class="container">
  <h2>Basic List Group</h2>
  <ul class="list-group">
    <li class="list-group-item">First item</li>
    <li class="list-group-item">Second item</li>
    <li class="list-group-item">Third item</li>
  </ul>
</div>
```

Basic List Group

First item

Second item

Third item

- Nếu muốn List Groups hiển thị chiều ngang, ta thêm lớp `.list-group-horizontal` vào lớp `.list-group`.

```
<ul class="list-group list-group-horizontal">
  <li class="list-group-item">First item</li>
  <li class="list-group-item">Second item</li>
  <li class="list-group-item">Third item</li>
  <li class="list-group-item">Fourth item</li>
</ul>
```

First item

Second item

Third item

Fourth item

- f) **Thẻ Card:** là một hộp có viền với một số phần đệm xung quanh nội dung của nó. Nó bao gồm các tùy chọn cho tiêu đề đầu trang (Header), tiêu đề cuối trang (Footer), nội dung, màu sắc, ...

```
<div class="container">
  <h2>Card Header and Footer</h2>
  <div class="card">
    <div class="card-header">Header</div>
    <div class="card-body">Content</div>
    <div class="card-footer">Footer</div>
  </div>
</div>
```

Card Header and Footer

Header

Content

Footer

- Sử dụng lớp `.card-title` để thêm tiêu đề thẻ vào bất kỳ yếu tố tiêu đề nào.
- Lớp `.card-link` thêm màu xanh lam vào bất kỳ liên kết nào và hiệu ứng chuột.

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ

```
<div class="card">
  <div class="card-body">
    <h4 class="card-title">Card title</h4>
    <p class="card-text">Some example text. Some example text.</p>
    <a href="#" class="card-link">Card link</a>
    <a href="#" class="card-link">Another link</a>
  </div>
</div>
```

Card title

Some example text. Some example text.

[Card link](#) [Another link](#)

g) Lớp Nav:

- Tạo thanh menu theo chiều ngang: sử dụng thẻ `` với lớp `.nav`, với mỗi phần tử trong thanh menu sử dụng thẻ `` với lớp `.nav-link`.

```
<ul class="nav">
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

[Link](#) [Link](#) [Link](#) Disabled

- Sử dụng lớp `.flex-column` tạo thanh menu theo chiều dọc.

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ

```
<ul class="nav flex-column">
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

Link

Link

Link

Disabled

- h) **Thẻ Form:** xác định một biểu mẫu được sử dụng để nhập dữ liệu của người dùng. Lớp `.form-group` để nhóm các phần tử trên form thành 1 nhóm hiển thị theo chiều dọc.

```
<div class="container">
  <h2>Stacked form</h2>
  <form action="/action_page.php">
    <div class="form-group">
      <label for="email">Email:</label>
      <input type="email" class="form-control" id="email" placeholder="Enter email" name="email">
    </div>
    <div class="form-group">
      <label for="pwd">Password:</label>
      <input type="password" class="form-control" id="pwd" placeholder="Enter password" name="pswd">
    </div>
    <div class="form-group form-check">
      <label class="form-check-label">
        <input class="form-check-input" type="checkbox" name="remember"> Remember me
      </label>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
  </form>
</div>
```

Stacked form

Email:

Password:

☐ Remember me

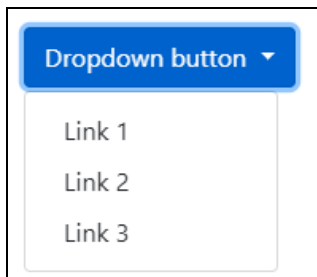
Sử dụng lớp .row tạo các dòng và lớp .col tạo các cột trên form.

```
<form action="/action_page.php">
  <div class="row">
    <div class="col">
      <input type="text" class="form-control" id="email" placeholder="Enter email" name="email">
    </div>
    <div class="col">
      <input type="password" class="form-control" placeholder="Enter password" name="pswd">
    </div>
  </div>
  <button type="submit" class="btn btn-primary mt-3">Submit</button>
</form>
```

i) Dropdown Menu:

- Lớp .dropdown dùng để tạo Dropdown menu.
- Lớp .dropdown-toggle và .dropdown-toggle="dropdown" tạo nút thả xuống.
- Lớp .dropdown-menu và .dropdown-item tạo các phần tử trong danh sách lựa chọn.

```
<div class="dropdown">
  <button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">
    Dropdown button
  </button>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">Link 1</a>
    <a class="dropdown-item" href="#">Link 2</a>
    <a class="dropdown-item" href="#">Link 3</a>
  </div>
</div>
```



2/ HtmlHelper:

- Đối tượng Html của lớp HtmlHelper cung cấp các phương thức tạo ra các thẻ HTML.
- HtmlHelper liên kết đối tượng Model với các thẻ HTML để hiển thị giá trị các thuộc Model lên các thẻ HTML và cũng gán giá trị của các thẻ HTML cho các thuộc tính Model sau khi submit biểu mẫu.
- Để làm việc với HtmlHelper, ta sử dụng namespace System.Web.Mvc.Html.
- Cấu trúc HtmlHelper :

HtmlHelper	Strogly Typed HtmlHelpers	Html Control
Html.BeginForm		Form
Html.ActionLink		Anchor link
Html.TextBox	Html.TextBoxFor	Textbox
Html.TextArea	Html.TextAreaFor	TextArea
Html.CheckBox	Html.CheckBoxFor	Checkbox
Html.RadioButton	Html.RadioButtonFor	Radio button
Html.DropDownList	Html.DropDownListFor	Dropdown, combobox
Html.ListBox	Html.ListBoxFor	multi-select list box
Html.Hidden	Html.HiddenFor	Hidden field
Password	Html.PasswordFor	Password textbox
Html.Display	Html.DisplayFor	Html text
Html.Label	Html.LabelFor	Label
Html.Editor	Html.EditorFor	Phát sinh Html controls tương ứng cho kiểu dữ liệu được xác định dựa vào thuộc tính Model.

Ví dụ:

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ

```
public class Student
{
    public int StudentId { get; set; }
    [Display(Name="Name")]
    public string StudentName { get; set; }
    public int Age { get; set; }
    public bool isNewlyEnrolled { get; set; }
    public string Password { get; set; }
    public DateTime DoB { get; set; }
}
```

```
StudentId: @Html.Editor("StudentId")
Student Name: @Html.Editor("StudentName")
Age: @Html.Editor("Age")
Password: @Html.Editor("Password")
isNewlyEnrolled: @Html.Editor("isNewlyEnrolled")
Gender: @Html.Editor("Gender")
DoB: @Html.Editor("DoB")
```

Kết quả hiển thị :

StudentId:	<input type="text" value="1"/>
Student Name:	<input type="text" value="Jogn"/>
Age:	<input type="text" value="19"/>
Password:	<input type="text" value="sdf"/>
isNewlyEnrolled:	<input checked="" type="checkbox"/>
Gender:	<input type="text" value="Boy"/>
DoB:	<input type="text" value="02-06-2015 11:39:15"/>

III. NỘI DUNG THỰC HÀNH BÀI 3:

1/ Chương trình mẫu:

- SV hãy tạo project và tên project là QLBH và thêm ADO.NET Entity Data Model là QLBHDBContext (tương tự bài 1).
- Thực hiện chức năng hiển thị danh sách nhân viên:
 - Trong **nhanvien.cs**, thêm các tên hiển thị cho các field (lưu ý: phải khai báo thêm **using System.ComponentModel**):

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ

```
[Key]
[StringLength(50), DisplayName("Mã nhân viên")]
public string manv { get; set; }

[StringLength(50), DisplayName("Tên nhân viên")]
public string tennv { get; set; }
[DisplayName("Ngày sinh"), DisplayFormat(DataFormatString = "{0:d}")]
public DateTime? ngaysinh { get; set; }
[DisplayName("Phái")]
public bool? phai { get; set; }

[StringLength(50), DisplayName("Địa chỉ")]
public string diachi { get; set; }

[StringLength(50), DisplayName("Password")]
public string password { get; set; }
```

- b. Thêm vào Controller cho bảng nhanvien với tên là nhanvienController và viết thêm code vào:

```
namespace QL BH.Controllers
{
    0 references
    public class nhanvienController : Controller
    {
        private QL BHDbContext db = new QL BHDbContext();

        0 references
        public ActionResult indexNV()
        {
            return View(db.nhanvien);
        }
    }
}
```

- c. Thêm view indexNV cho nhanvien.
d. Trong trang indexNV.cshtml, SV hãy nhập lại :

```
@model IEnumerable<QL BH.Models.nhanvien>
@{
    Layout = null;
}
<!DOCTYPE html>
```

```
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap-4.5.0-dist/css/bootstrap.min.css"
rel="stylesheet" />
    <script src="~/Content/bootstrap-4.5.0-
dist/js/bootstrap.min.js"></script>
    <title>Index</title>
</head>
<body>
    <h2>Quản lý nhân viên</h2>
    <p>
        @Html.ActionLink("Thêm mới", "themNV")
    </p>
    <table class="table">
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.tennv)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.ngaysinh)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.phai)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.diachi)
            </th>
            <th></th>
        </tr>

        @foreach (var item in Model)
        {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.tennv)
                </td>
                <td>
                    @*@Html.DisplayFor(modelItem => item.ngaysinh)*@
                    @item.ngaysinh.Value.ToShortDateString()
                </td>
                <td>
                    @{
                        if (item.phai == true)
                        {
                            <span>Nam</span>
                        }
                        else
                        {

```

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ

```
                <span>Nữ</span>
            }
        }
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.diachi)
    </td>
    <td>
        @Html.ActionLink("Sửa", "suaNV", new { id = item.manv })
        @Html.ActionLink("Xóa", "xoaNV", new { id = item.manv })
    </td>
</tr>
}
</table>
</body>
</html>
```

e. Kết quả thực thi trang indexNV.cshtml :

Quản lý nhân viên				
Thêm mới				
Tên nhân viên	Ngày sinh	Phái	Địa chỉ	
Nguyễn Văn A	01/03/1980	Nam	120 Trần Hưng Đạo	Sửa Xóa
Trần Thị B	02/04/1981	Nữ	200 Lý Thái Tổ	Sửa Xóa

- c) Thực hiện chức năng thêm nhân viên:
- Thêm view **themNV** cho nhanvien.
 - Viết code cho trang **themNV.cshtml**:

```
@model QBH.Models.nhanvien
@{
    Layout = null;
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap-4.5.0-dist/css/bootstrap.min.css"
rel="stylesheet" />
    <script src="~/Content/bootstrap-4.5.0-
dist/js/bootstrap.min.js"></script>
    <title>Create</title>
</head>
<body>
```

```
@using (Html.BeginForm("themNV", "nhanvien", FormMethod.Post))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Thêm nhân viên</h4>
        <div>
            @Html.ActionLink("Quay lại", "indexNV")
        </div>
        <hr />
        <div class="form-group">
            @Html.LabelFor(model => model.manv, htmlAttributes: new {
@class = "control-label col-md-2" })
            <div class="col-md-6">
                @Html.EditorFor(model => model.manv, new { htmlAttributes
= new { @class = "form-control" } })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.tennv, htmlAttributes: new {
@class = "control-label col-md-2" })
            <div class="col-md-6">
                @Html.EditorFor(model => model.tennv, new {
htmlAttributes = new { @class = "form-control" } })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.ngaysinh, htmlAttributes: new {
@class = "control-label col-md-2" })
            <div class="col-md-6">
                @Html.EditorFor(model => model.ngaysinh, new {
htmlAttributes = new { @class = "form-control", @type = "date" } })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.phai, htmlAttributes: new {
@class = "control-label col-md-2" })
            <div class="col-md-6">
                @Html.DropDownListFor(model=>model.phai, new
List<SelectListItem> {
                    new SelectListItem { Text="Nam",Value="true"},
                    new SelectListItem { Text="Nữ",Value="false" } },
                    htmlAttributes: new { @class = "custom-select" }
                )
            </div>
        </div>
    </div>
}
```

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ

```
        <div class="form-group">
            @Html.LabelFor(model => model.diachi, htmlAttributes: new {
@class = "control-label col-md-2" })
            <div class="col-md-6">
                @Html.EditorFor(model => model.diachi, new {
htmlAttributes = new { @class = "form-control" } })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-6">
                <input type="submit" value="Thêm" class="btn btn-primary"
/>
            </div>
        </div>
    </div>
}
</body>
</html>
```

c. Trong nhanvienController, ta thêm code :

```
public ActionResult themNV()
{
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult themNV([Bind(Include =
"manv,tennv,ngaysinh,phai,diachi,password")]
    nhanvien nhanvien)
{
    db.nhanvien.Add(nhanvien);
    db.SaveChanges();
    return RedirectToAction("indexNV");
}
```


d. Kết quả khi thêm nhân viên:

Thêm nhân viên
[Quay lại](#)

Mã nhân viên


Tên nhân viên

Ngày sinh



Phái

Nam



Địa chỉ

Thêm

2/ Bài tập tổng hợp:

- SV hãy thực hiện chức năng xóa nhân viên.
- SV hãy thực hiện chức năng sửa thông tin nhân viên.

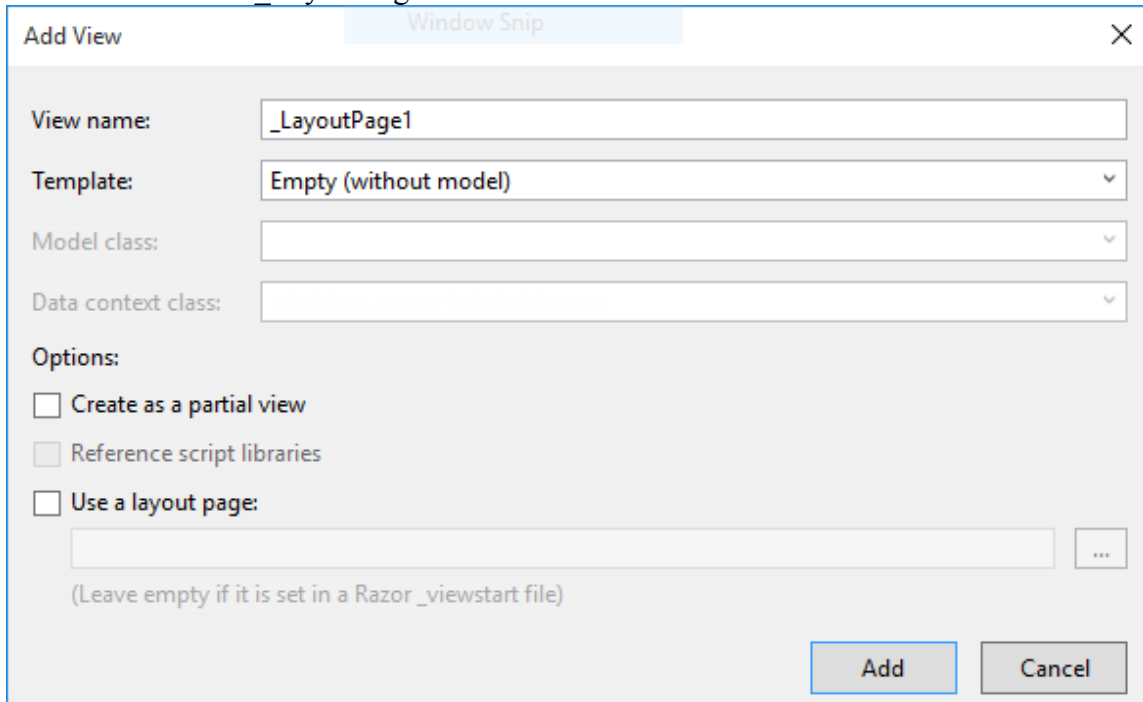
3/ Bài tập làm thêm:

SV hãy thực hiện chức năng Quản lý khách hàng với các chức năng hiển thị danh sách khách hàng, thêm, xóa, sửa thông tin khách hàng.

IV. NỘI DUNG THỰC HÀNH BÀI 4:

1/ Chương trình mẫu:

- **Thêm layout vào project:**
 - Click phải vào thư mục Views ⇒ Add ⇒ chọn View.
 - Thêm view `_LayoutPage1` như hình.



- Nhấn Add, ta tạo được view `_LayoutPage1`, thêm Bootstrap vào view, ta có view như sau:

```
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap-4.5.0-dist/css/bootstrap.min.css" rel="stylesheet" />
    <script src="~/Content/bootstrap-4.5.0-dist/js/bootstrap.min.js"></script>
</head>
<body>
    <div>
    </div>
</body>
</html>
```

- **Thiết kế layout:**
 - Ta thiết kế layout như sau :

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ

Header

Phần Header

Trang chủ

Trang chủ

Trang chủ

Phần Title

Trang chủ

Trang chủ

Trang chủ

Quản lý Nhà sản xuất

Quản lý Nhân viên

Quản lý Khách hàng

Quản lý Loại hàng hóa

Quản lý Hàng hóa

Phần Body

Quản lý nhân viên

Thêm mới

Tên nhân viên	Ngày sinh	Phái	Địa chỉ	
Nguyễn Văn A	01/03/1980	Nam	120 Trần Hưng Đạo	Sửa Xóa
Trần Thị B	02/04/1981	Nữ	200 Lý Thái Tổ	Sửa Xóa

Footer

Phần Footer

- Code cho _LayoutPage1 :

```
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap-4.5.0-dist/css/bootstrap.min.css"
rel="stylesheet" />
    <script src="~/Content/bootstrap-4.5.0-
dist/js/bootstrap.min.js"></script>
</head>
<body>
    <div class="container-fluid">
        <div class="card">
            <div class="card-header bg-secondary text-white">
                Header
            </div>
            <div class="card-title">
                <div class="row">
                    <div class="col-8">
                        <div class="nav nav-tabs">
                            <div class="nav-item bg-primary">
                                <div class="nav-link">
                                    <a href="@Url.Action("Index","Home")"
class="text-white">Trang chủ</a>
                                </div>
                            </div>
                            <div class="nav-item bg-primary">
                                <div class="nav-link">
                                    @Html.ActionLink("Trang chủ", "Index",
"Home", null, new { @class = "text-white" })
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
```



```
        </div>
    </div>
    <div class="nav-item bg-primary">
        <div class="nav-link">
            @Html.ActionLink("Trang chủ", "Index",
"Home", null, new { @class = "text-white" })
        </div>
    </div>
</div>

<div class="col-4">
    <div class="nav nav-tabs float-right">
        <div class="nav-item bg-primary">
            <div class="nav-link">
                @Html.ActionLink("Trang chủ", "Index",
"Home", null, new { @class = "text-white" })
            </div>
        </div>
        <div class="nav-item bg-primary">
            <div class="nav-link">
                @Html.ActionLink("Trang chủ", "Index",
"Home", null, new { @class = "text-white" })
            </div>
        </div>
    </div>
</div>
</div>
</div>
<div class="card-body">
    @RenderBody()
</div>
<div class="card-footer bg-secondary text-white">
    Footer
</div>
</div>
</div>
</body>
</html>
```

- Sử dụng layout cho các view:

Các view muốn sử dụng layout, ta khai báo thêm layout như sau:

Bài 3, 4: ASP.NET MVC, BOOTSTRAP, HTMLHELPER VÀ TRUY XUẤT CSDL BẰNG LINQ

```
@model IEnumerable<QLBH.Models.nhanvien>
@{
    Layout = "~/Views/_LayoutPage1.cshtml";
}
<!DOCTYPE html>
<html>
<html>...</html>
<body>...</body>
</html>
```

2/ Bài tập tổng hợp:

SV hãy viết code cho phần body của layout như trong hình:

Header

Phần Header

Trang chủ Trang chủ Trang chủ

Phần Title

Trang chủ Trang chủ Trang chủ

Phần Body

Quản lý Nhà sản xuất

Quản lý Nhân viên

Quản lý Khách hàng

Quản lý Loại hàng hóa

Quản lý Hàng hóa

Quản lý nhân viên

Thêm mới

Tên nhân viên	Ngày sinh	Phái	Địa chỉ	
Nguyễn Văn A	01/03/1980	Nam	120 Trần Hưng Đạo	Sửa Xóa
Trần Thị B	02/04/1981	Nữ	200 Lý Thái Tổ	Sửa Xóa

Footer

Phần Footer

- Khi chọn mục “Quản lý Nhà sản xuất” thì thực hiện chức năng “Quản lý Nhà sản xuất” như ở Bài 1, 2.
- Khi chọn mục “Quản lý Nhân viên” thì thực hiện chức năng “Quản lý Nhân viên” như ở Bài 3.
- Khi chọn mục “Quản lý Loại hàng hóa” thì thực hiện chức năng “Quản lý Loại hàng hóa”. SV hãy thực hiện thêm chức năng quản lý loại hàng hóa.

3/ Bài tập làm thêm:

- Khi chọn mục “Quản lý Khách hàng” thì thực hiện chức năng “Quản lý Khách hàng”. SV hãy thực hiện thêm chức năng quản lý khách hàng.
- Khi chọn mục “Quản lý Hàng hóa” thì thực hiện chức năng “Quản lý Hàng hóa”. SV hãy thực hiện thêm chức năng quản lý hàng hóa.

V. MỤC TIÊU:

Sau khi thực hành bài này, SV có thể :

- Hiểu và vận dụng Validation.
- Đối tượng Session.
- Thao tác trên 1 Table trong CSDL.
- Thực hiện các phép toán hiển thị danh sách, thêm, xóa, sửa.

VI. TÓM TẮT LÝ THUYẾT:

1/ **Validation** : dùng để kiểm tra dữ liệu.

a. Code từ Model : sử dụng namespace **System.ComponentModel.DataAnnotations**.

Bảng các thuộc tính dữ liệu :

Thuộc tính	Giải thích
Required	Chỉ định thuộc tính phải có dữ liệu nhập vào trước khi submit về server.
StringLength	Định nghĩa chiều dài thuộc tính, cho phép đặc tả cả chiều dài tối thiểu và tối đa.
Range	Định nghĩa giá trị số tối thiểu và tối đa của một thuộc tính.
RegularExpression	Là biểu thức chính quy được dùng để xử lý chuỗi nâng cao thông qua biểu thức riêng của nó
CreditCard	Định nghĩa số thẻ tính dụng (credit card)
CustomValidation	Cho phép người dùng tự viết hàm xử lý kiểm tra lỗi
EmailAddress	Kiểm tra email người dùng nhập vào có hợp lệ?
FileExtension	Kiểm tra phần mở rộng của một tập tin
MaxLength	Kích thước tối đa của trường dữ liệu nhập vào
MinLength	Kích thước tối thiểu của trường dữ liệu nhập vào
Phone	Kiểm tra số phone có hợp lệ?

b. Code từ View :

- Sử dụng thư viện **System.Web.Mvc.Html** để xác định các phương thức:

Phương thức	Mô tả
ValidationMessage ValidationMessageFor	Hiển thị thông báo xác thực nếu có lỗi trong mục nhập được chỉ định trong đối tượng ModelStateDictionary.
ValidationSummary	Tạo một danh sách không có thứ tự các thông báo xác thực trong đối tượng ModelStateDictionary và được sử dụng hiển thị tất cả các thông báo lỗi cho các trường.

- Sử dụng thư viện **System.Web.Mvc** để xác định các phương thức:

Phương thức HtmlHelper.AntiForgeryToken	Tạo một trường ẩn trên Form khi Form được submit.
Lớp đối tượng ValidationAntiForgeryTokenAttribute	Biểu diễn một thuộc tính nhằm ngăn chặn giả mạo theo yêu cầu.

2/ Đối tượng Session

- Session là một vùng nhớ tồn tại trong một phiên làm việc giữa client và server, bắt đầu khi có yêu cầu từ client và kết thúc khi người dùng giải phóng vùng nhớ Session hoặc khi Session hết thời gian timeout.
- Session được dùng để lưu thông tin liên quan đến người dùng nhưng thay vì được lưu trên client, các biến session được lưu trên server.
- Mỗi Session có một định danh duy nhất (SessionID).

VII. NỘI DUNG THỰC HÀNH BÀI 5:

1/ Chương trình mẫu:

- Thêm các ràng buộc dữ liệu trong model **nhanvien.cs**:

```
[Key]
[StringLength(50), DisplayName("Mã nhân viên")]
[Required(ErrorMessage = "Xin nhập mã nhân viên!")]
5 references
public string manv { get; set; }
[StringLength(50), DisplayName("Tên nhân viên")]
[Required(ErrorMessage = "Xin nhập tên nhân viên!")]
6 references
public string tennv { get; set; }
[DisplayName("Ngày sinh"), DisplayFormat(DataFormatString = "{0:d}")]
[Required(ErrorMessage = "Xin nhập ngày sinh!")]
6 references
public DateTime? ngaysinh { get; set; }
[DisplayName("Phái")]
6 references
public bool? phai { get; set; }

[StringLength(50), DisplayName("Địa chỉ")]
[Required(ErrorMessage = "Xin nhập địa chỉ!")]
6 references
public string diachi { get; set; }
```

- Thêm các kiểm tra ràng buộc trong view **themNV** :

```
@model QBH.Models.nhanvien
@{
    Layout = null;
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap-4.5.0-dist/css/bootstrap.min.css"
rel="stylesheet" />
    <script src="~/Content/bootstrap-4.5.0-
dist/js/bootstrap.min.js"></script>
    <title>Create</title>
</head>
<body>
    @using (Html.BeginForm("themNV", "nhanvien", FormMethod.Post))
    {
        @Html.AntiForgeryToken()

        <div class="form-horizontal">
            <h4>Thêm nhân viên</h4>
            <div>
                @Html.ActionLink("Quay lại", "indexNV")
            </div>
        </div>
    }
</body>
</html>
```

```

        <div class="form-group">
            @Html.LabelFor(model => model.manv, htmlAttributes: new { @class
= "control-label col-md-2" })
            <div class="col-md-6">
                @Html.EditorFor(model => model.manv, new { htmlAttributes =
new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.manv, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.tennv, htmlAttributes: new { @class
= "control-label col-md-2" })
            <div class="col-md-6">
                @Html.EditorFor(model => model.tennv, new { htmlAttributes =
new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.tennv, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.ngaysinh, htmlAttributes: new {
@class = "control-label col-md-2" })
            <div class="col-md-6">
                @Html.EditorFor(model => model.ngaysinh, new { htmlAttributes
= new { @class = "form-control", @type = "date" } })
                @Html.ValidationMessageFor(model => model.ngaysinh, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.phai, htmlAttributes: new { @class
= "control-label col-md-2" })
            <div class="col-md-6">
                @Html.DropDownListFor(model=>model.phai,new
List<SelectListItem> {
                    new SelectListItem { Text="Nam",Value="true"},
                    new SelectListItem { Text="Nữ",Value="Nữ" } },
                    htmlAttributes: new { @class = "custom-select" }
                )
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.diachi, htmlAttributes: new {
@class = "control-label col-md-2" })

```

```

        <div class="col-md-6">
            @Html.EditorFor(model => model.diachi, new { htmlAttributes =
new { @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.diachi, "", new {
@class = "text-danger" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-6">
            <input type="submit" value="Thêm" class="btn btn-primary" />
        </div>
    </div>
</div>
}
</body>
</html>

```

- Trong controller **nhanvienController**, ta thêm lệnh kiểm tra trạng thái Model:

```

public ActionResult themNV()
{
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult themNV([Bind(Include =
"manv,tennv,ngaysinh,phai,diachi,password")]
    nhanvien nhanvien)
{
    if (ModelState.IsValid)
    {
        db.nhanvien.Add(nhanvien);
        db.SaveChanges();
        return RedirectToAction("indexNV");
    }
    return View(nhanvien);
}

```

- Thi thực thi, ta sẽ thấy rằng buộc dữ liệu khi thêm nhân viên :

Thêm nhân viên

[Quay lại](#)

Mã nhân viên

Xin nhập mã nhân viên!

Tên nhân viên

Xin nhập tên nhân viên!

Ngày sinh

dd/mm/yyyy

Xin nhập ngày sinh!

Phái

Nam

Địa chỉ

Xin nhập địa chỉ!

Thêm

2/ Bài tập tổng hợp:

- Chức năng “Xóa nhân viên”: Viết đoạn chương trình kiểm tra ràng buộc khóa ngoại (nhân viên chỉ xóa được khi chưa có mối quan hệ với mẫu tin trong bảng “Phiếu giao hàng”).
 - Khi Nhân viên chọn xóa chưa có mối quan hệ với mẫu tin trong bảng “Phiếu giao hàng” thì hiển thị View thông báo như hình 5.1 và cho phép người dùng xóa Nhân viên trong CSDL. Ngược lại, hiển thị View thông báo như hình 5.2 không cho phép người dùng xóa Nhân viên.

Bạn muốn xóa nhân viên này?

[Quay lại](#)

Tên nhân viên
Trần Thị B

Ngày sinh
02/04/1981

Phái
Nữ

Địa chỉ
200 Lý Thái Tổ

[Xóa](#)

Hình 5.1

Bạn không xóa được nhân viên này!

[Quay lại](#)

Tên nhân viên
Nguyễn Văn A

Ngày sinh
01/03/1980

Phái
Nam

Địa chỉ
120 Trần Hưng Đạo

Hình 5.2

- b. Chức năng “Thêm nhân viên”: viết đoạn chương trình kiểm tra ràng buộc khóa chính (Mã số nhân viên là duy nhất).
- Nếu người dùng nhập Mã số nhân viên đã tồn tại trong hệ thống thì hiển thị View thông báo lỗi như hình 5.3.

Nhân viên này đã tồn tại trong hệ thống!

Thêm nhân viên

Mã nhân viên
NV001

Tên nhân viên
Đinh Văn Lành

Ngày sinh
11/02/1989

Phái
Nam

Địa chỉ
45 Hoàng Văn Thụ

[Quay lại](#)

Hình 5.3


- Khi người dùng nhấn mục “Quay lại” trong view Hình 5.3 thì hiển thị view “Thêm nhân viên” như hình 5.4.

Thêm nhân viên

[Quay lại](#)

Mã nhân viên

Tên nhân viên

Ngày sinh
 

Phái

Địa chỉ

Hình 5.4

V. NỘI DUNG THỰC HÀNH BÀI 6:

1/ Chương trình mẫu: thực hiện chức năng đăng nhập khách hàng

- Thêm vào Controller cho bảng **khachhang** với tên là **khachhangController** và viết thêm code vào :

```
public class khachhangController : Controller
{
    private QLBDHDBContext db = new QLBDHDBContext();
    // GET: khachhang
    public ActionResult indexDangNhap()
    {
        ViewBag.kiemtraDangnhap = null;
        return View();
    }
}
```

- Thêm view **indexDangNhap** cho **khachhang**
- Trong trang **indexDangNhap.cshtml**, SV hãy nhập lại :

```
@model QLBDH.Models.khachhang

@{
    ViewBag.Title = "Đăng nhập";
    Layout = "~/Views/_LayoutPage1.cshtml";
}

<h2>Đăng nhập hệ thống</h2>
@using (Html.BeginForm("dangnhap", "khachhang", FormMethod.Post))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            <div class="control-label">
                <span class="font-weight-bold">Tên tài khoản</span>
            </div>
            <div class="col-md-6">
                @Html.EditorFor(model => model.makh, new { htmlAttributes
= new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.makh, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="control-label">
```

```

        <span class="font-weight-bold">Mật khẩu</span>
    </div>
    <div class="col-md-3">
        @Html.PasswordFor(model => model.password, new {
htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.password, "",
new { @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-6">
        <input type="submit" value="Đăng nhập" class="btn btn-
primary" />
    </div>
</div>
@if (ViewBag.kiemtraDangnhap != null &&
(bool)ViewBag.kiemtraDangnhap == false)
{
    <div class="form-group">
        <div class="control-label col-md-12">
            <span class="font-weight-bold text-danger">Bạn nhập
"Tên tài khoản" hay "Mật khẩu" bị sai!</span>
        </div>
    </div>
}
</div>
}

<div>
    @Html.ActionLink("Trở về", "indexNV", "nhanvien")
</div>

```

d) Code cho trang **_LayoutPage1.cshtml** :

```

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap-4.5.0-dist/css/bootstrap.min.css"
rel="stylesheet" />
    <script src="~/Content/bootstrap-4.5.0-
dist/js/bootstrap.min.js"></script>
</head>
<body>
    <div class="container-fluid">
        <div class="card">
            <div class="card-header bg-secondary text-white">

```

```

        Header
    </div>
    <div class="card-title">
        <div class="row">
            <div class="col-8">
                <div class="nav nav-tabs">
                    <div class="nav-item bg-primary">
                        <div class="nav-link">
                            <a href="@Url.Action("Index","Home")"
class="text-white">Trang chủ</a>
                        </div>
                    </div>
                    <div class="nav-item bg-primary">
                        <div class="nav-link">
                            @Html.ActionLink("Trang chủ",
"Index","Home", null, new { @class = "text-white" })
                        </div>
                    </div>
                    <div class="nav-item bg-primary">
                        <div class="nav-link">
                            @Html.ActionLink("Trang chủ",
"Index","Home", null, new { @class = "text-white" })
                        </div>
                    </div>
                </div>
            </div>
            <div class="col-4">
                <div class="nav nav-tabs float-right">
                    <div class="nav-item bg-primary">
                        <div class="nav-link">
                            @Html.ActionLink("Trang chủ",
"Index","Home", null, new { @class = "text-white" })
                        </div>
                    </div>
                    <div class="nav-item bg-primary">
                        <div class="nav-link">
                            @if (Session["dangnhap"] == null)
                            {
                                <a
href="@Url.Action("indexDangNhap","khachhang")" class="text-white">Đăng
nhập</a>
                            }
                            else
                            {
                                <a
href="@Url.Action("dangxuat","khachhang")" class="text-white">Đăng xuất</a>
                            }
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
        <div class="nav-item bg-primary">
            <div class="nav-link">
                @if (Session["dangnhap"] == null)
                {
                    <a
href="@Url.Action("dangky","khachhang")" class="text-white">Đăng ký</a>
                }
                else
                {
                    QL BH.Models.khachhang kh =
Session["dangnhap"] as QL BH.Models.khachhang;
                    <span>@kh.tenkh</span>
                }
            </div>
        </div>
    </div>
</div>
</div>
</div>
<div class="card-body">
    <div class="row">
        <div class="col-md-3">
            <div class="list-group">
                <div class="list-group-item bg-info ">
                    <a
href="@Url.Action("indexNSX","nhasanxuat")" class="text-white">Quản lý Nhà
sản xuất</a>
                </div>
                <div class="list-group-item bg-info ">
                    <a href="@Url.Action("indexNV","nhanvien")"
class="text-white">Quản lý Nhân viên</a>
                </div>
                <div class="list-group-item bg-info ">
                    <a href="@Url.Action("indexKH","khachhang")"
class="text-white">Quản lý Khách hàng</a>
                </div>
                <div class="list-group-item bg-info ">
                    <a
href="@Url.Action("indexLHH","loaihanghoa")" class="text-white">Quản lý Loại
hàng hóa</a>
                </div>
                <div class="list-group-item bg-info ">
                    <a href="@Url.Action("indexHH","hanghoa")"
class="text-white">Quản lý Hàng hóa</a>
                </div>
            </div>
        </div>
        <div class="col-md-9 bg-light">

```

```

        @RenderBody()
    </div>
</div>
</div>
<div class="card-footer bg-secondary text-white">
    Footer
</div>
</div>
</div>
</body>
</html>

```

e) Kết quả khi ta thực thi sẽ có hiển thị chức năng đăng nhập, đăng ký :

Header

Trang chủ Trang chủ Trang chủ

Trang chủ Đăng nhập Đăng ký

Quản lý Nhà sản xuất

Quản lý Nhân viên

Quản lý Khách hàng

Quản lý Loại hàng hóa

Quản lý Hàng hóa

Quản lý nhân viên

Thêm mới

Tên nhân viên	Ngày sinh	Phái	Địa chỉ	
Nguyễn Văn A	01/03/1980	Nam	120 Trần Hưng Đạo	Sửa Xóa
Trần Thị B	02/04/1981	Nữ	200 Lý Thái Tổ	Sửa Xóa

Footer

f) Khi nhấn vào Đăng nhập sẽ xuất hiện trang đăng nhập :

Header

Trang chủ Trang chủ Trang chủ

Trang chủ Đăng nhập Đăng ký

Quản lý Nhà sản xuất

Quản lý Nhân viên

Quản lý Khách hàng

Quản lý Loại hàng hóa

Quản lý Hàng hóa

Đăng nhập hệ thống

Tên tài khoản

Mật khẩu

Đăng nhập

Trở về

Footer

g) Khi đăng nhập thành công sẽ xuất hiện chức năng đăng xuất:

Header

[Trang chủ](#)
[Trang chủ](#)
[Trang chủ](#)

[Trang chủ](#)
[Đăng xuất](#)
Trần Văn Giàu

Quản lý Nhà sản xuất

Quản lý Nhân viên

Quản lý Khách hàng

Quản lý Loại hàng hóa

Quản lý Hàng hóa

Quản lý nhân viên

[Thêm mới](#)

Tên nhân viên	Ngày sinh	Phái	Địa chỉ	
Nguyễn Văn A	01/03/1980	Nam	120 Trần Hưng Đạo	Sửa Xóa
Trần Thị B	02/04/1981	Nữ	200 Lý Thái Tổ	Sửa Xóa

Footer

h) Đăng nhập thất bại :

Header

[Trang chủ](#)
[Trang chủ](#)
[Trang chủ](#)

[Trang chủ](#)
[Đăng nhập](#)
[Đăng ký](#)

Quản lý Nhà sản xuất

Quản lý Nhân viên

Quản lý Khách hàng

Quản lý Loại hàng hóa

Quản lý Hàng hóa

Đăng nhập hệ thống

Tên tài khoản

Mật khẩu

[Đăng nhập](#)

Bạn nhập "Tên tài khoản" hay "Mật khẩu" bị sai!

[Trở về](#)

Footer

2/ Bài tập tổng hợp:

a) SV hãy thực hiện chức năng “Đăng xuất”: khi nhấn “Đăng xuất” như hình :

Header

Trang chủ
Trang chủ
Trang chủ
Trang chủ
Đăng xuất
Trần Văn Giàu

Quản lý Nhà sản xuất
Quản lý Nhân viên
Quản lý Khách hàng
Quản lý Loại hàng hóa
Quản lý Hàng hóa

Quản lý nhân viên

[Thêm mới](#)

Tên nhân viên	Ngày sinh	Phái	Địa chỉ	
Nguyễn Văn A	01/03/1980	Nam	120 Trần Hưng Đạo	Sửa Xóa
Trần Thị B	02/04/1981	Nữ	200 Lý Thái Tổ	Sửa Xóa

Footer

Sẽ trở về trang quản lý nhân viên :

Header

Trang chủ
Trang chủ
Trang chủ
Trang chủ
Đăng nhập
Đăng ký

Quản lý Nhà sản xuất
Quản lý Nhân viên
Quản lý Khách hàng
Quản lý Loại hàng hóa
Quản lý Hàng hóa

Quản lý nhân viên

[Thêm mới](#)

Tên nhân viên	Ngày sinh	Phái	Địa chỉ	
Nguyễn Văn A	01/03/1980	Nam	120 Trần Hưng Đạo	Sửa Xóa
Trần Thị B	02/04/1981	Nữ	200 Lý Thái Tổ	Sửa Xóa

Footer

- b) SV hãy thực hiện chức năng thay đổi thông tin khách hàng: khi nhấn vào tên khách hàng như trong hình :

Header

Trang chủ
Trang chủ
Trang chủ
Trang chủ
Đăng xuất
Trần Văn Giàu

Quản lý Nhà sản xuất
Quản lý Nhân viên
Quản lý Khách hàng
Quản lý Loại hàng hóa
Quản lý Hàng hóa

Quản lý nhân viên

[Thêm mới](#)

Tên nhân viên	Ngày sinh	Phái	Địa chỉ	
Nguyễn Văn A	01/03/1980	Nam	120 Trần Hưng Đạo	Sửa Xóa
Trần Thị B	02/04/1981	Nữ	200 Lý Thái Tổ	Sửa Xóa

Footer

Sẽ xuất hiện trang cho phép thay đổi thông tin khách hàng.

I. MỤC TIÊU:

Sau khi thực hành bài này, SV có thể hiểu và vận dụng:

- Partial View. Thẻ file, Image.
- JQuery và Ajax.
- Thao tác trên nhiều Table trong CSDL.
- Thực hiện các phép toán thêm, xóa, sửa, tìm kiếm, và các ràng buộc dữ liệu.

II. TÓM TẮT LÝ THUYẾT:

1/ Partial View:

- Công dụng: Partial view trong ASP.NET MVC là một view sử dụng lại, có thể được sử dụng như là một view con trong nhiều view khác. Partial view giúp loại bỏ trùng lặp mã bằng cách sử dụng lại cùng một partial view ở nhiều view khác. Chúng ta có thể sử dụng partial view là một phần trong layout view, cũng như nội dung view khác.
- Tạo một Partial View mới :
 - Click chuột phải lên thư mục Views => Add => View.
 - Trong hộp thoại Add View, nhập tên View trong mục “View name” và đánh dấu chọn “Create as a partial view” => nhấn Add.

The screenshot shows the 'Add View' dialog box. The 'View name' field contains the text '_TenCuaPartialView'. The 'Template' dropdown is set to 'Empty (without model)'. The 'Model class' and 'Data context class' fields are empty. In the 'Options' section, the checkbox 'Create as a partial view' is checked. Below it, the checkbox 'Reference script libraries' is unchecked, and the checkbox 'Use a layout page' is checked. The 'Layout page' field contains the path '~/Views/_LayoutPage1.cshtml'. At the bottom right, there are 'Add' and 'Cancel' buttons.

Bài 7, 8: ASP.NET MVC, JQUERY VÀ TRUY XUẤT CSDL BẰNG LINQ

c. Hiển thị Partial View: trong view cha, chúng ta có thể dùng các phương thức sau để hiển thị Partial View: `Html.Partial()`, `Html.RenderPartial()`, `Html.RenderAction()`.

2/ **Thẻ file**: dùng để chọn tập tin.

Ví dụ :

```
<input type="file" name="hinh" class="form-control-file border" />
```

3/ **Thẻ img**: dùng để hiển thị hình ảnh.

Ví dụ:

```

```

```

```

4/ **Jquery**: Jquery là thư viện mở của JavaScript

a. Khai báo Jquery:

- Download Jquery từ trang <https://jquery.com/download/>
- Khai báo sử dụng Jquery trong lớp View:

```
<!DOCTYPE html>

<html>
<head>
    <script src="~/Scripts/jquery-3.3.1.js"></script>
</head>
<body>
</body>
</html>
```

b. Jquery selector :

Selector	Miêu tả
Tag Name	Biểu diễn một tên thẻ có sẵn trong DOM. Ví dụ: <code>\$('p')</code> chọn tất cả đoạn văn trong phần tử.
Tag ID	Biểu diễn một tên thẻ có sẵn với ID đã cho trong DOM. Ví dụ <code>\$('#some-id')</code> chọn tất cả phần tử đơn trong tài liệu mà có một ID là some-id
Tag Class	Biểu diễn một thẻ có sẵn với lớp đã cho trong DOM. Ví

độ \$('some-class') chọn tất cả các phần tử trong tài liệu mà có một lớp là some-class.

- 5/ **Ajax JQuery**: là một bộ công cụ cho phép load dữ liệu từ server mà không yêu cầu tải lại trang. Các phương thức của Ajax JQuery :
- a. \$(selector).load(URL, data, callback) : cho phép load nội dung HTML hoặc nội dung văn bản được tải từ máy chủ và thêm vào thành phần DOM.
 - b. \$.get(URL, callback): gửi yêu cầu không đồng bộ HttpGet đến máy chủ và nhận phản hồi trở về.
 - c. \$.post(URL, data, callback): gửi yêu cầu không đồng bộ HttpPost đến máy chủ để gửi dữ liệu đến máy chủ và nhận được phản hồi trở về.

III.NỘI DUNG THỰC HÀNH BÀI 7:

- 1/ **Chương trình mẫu: thực hiện chức năng quản lý hàng hóa (sử dụng layout Bài 4)**
- a) Trong Model bảng “hanghoa”, SV hãy thêm vào:

Bài 7, 8: ASP.NET MVC, JQUERY VÀ TRUY XUẤT CSDL BẰNG LINQ

```
[Key]
[StringLength(50), DisplayName("Mã hàng hóa")]
[Required(ErrorMessage = "Xin nhập mã hàng hóa!")]
10 references
public string mahang { get; set; }

[StringLength(50), DisplayName("Tên hàng")]
[Required(ErrorMessage = "Xin nhập Tên hàng!")]
12 references
public string tenhang { get; set; }

[StringLength(50), DisplayName("Đơn vị tính")]
[Required(ErrorMessage = "Xin nhập Đơn vị tính!")]
12 references
public string donvitinh { get; set; }

[DisplayName("Đơn giá")]
[Required(ErrorMessage = "Xin nhập Đơn giá!")]
12 references
public double? dongia { get; set; }

[StringLength(50), DisplayName("Hình ảnh")]
10 references
public string hinh { get; set; }

[StringLength(50), DisplayName("Mã loại hàng")]
7 references
public string maloi { get; set; }

[StringLength(50), DisplayName("Mã nhà sản xuất")]
7 references
public string mansx { get; set; }
```

b) Trong Model bảng “loaihanghoa”, SV hãy thêm vào:

```
[Key]
[StringLength(50), DisplayName("Mã loại hàng")]
[Required(ErrorMessage = "Xin nhập mã loại hàng!")]
0 references
public string maloi { get; set; }

[StringLength(50), DisplayName("Tên loại hàng")]
[Required(ErrorMessage = "Xin nhập tên loại hàng!")]
6 references
public string tenloai { get; set; }
```

c) Trong Model bảng “nhasanxuat”, SV hãy thêm vào:

Bài 7, 8: ASP.NET MVC, JQUERY VÀ TRUY XUẤT CSDL BẰNG LINQ

```
[Key]
[StringLength(50), DisplayName("Mã nhà sản xuất")]
[Required(ErrorMessage = "Xin nhập mã nhà sản xuất!")]
1 reference
public string mansx { get; set; }

[StringLength(50), DisplayName("Tên nhà sản xuất")]
[Required(ErrorMessage = "Xin nhập tên nhà sản xuất!")]
7 references
public string tennsx { get; set; }

[StringLength(50), DisplayName("Địa chỉ")]
[Required(ErrorMessage = "Xin nhập địa chỉ!")]
1 reference
public string diachi { get; set; }
```

- d) Thêm vào Controller cho bảng **hanghoa** với tên là **hanghoaController** và viết thêm code vào :

```
public class hanghoaController : Controller
{
    private QLBDHDBContext db = new QLBDHDBContext();

    // GET: hanghoa
    0 references
    public ActionResult indexHH()
    {
        var hanghoa = db.hanghoa.Include(h => h.loaihanghoa).Include(h => h.nhasanxuat);
        return View(hanghoa.ToList());
    }
}
```

- e) Thêm View indexHH cho hanghoa. Trong trang indexHH.cshtml, SV hãy nhập lại:

```
@model IEnumerable<QLBH.Models.hanghoa>

@{
    ViewBag.Title = "Quản lý hàng hóa";
    Layout = "~/Views/_LayoutPage1.cshtml";
}

<h2>Quản lý hàng hóa</h2>

<p>
    @Html.ActionLink("Thêm hàng hóa", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.mahang)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.tenhang)
        </th>
    </tr>
</table>
```

```
</th>
<th>
    @Html.DisplayNameFor(model => model.donvitinh)
</th>
<th>
    @Html.DisplayNameFor(model => model.dongia)
</th>
<th>
    @Html.DisplayNameFor(model => model.loaihanghoa.tenloai)
</th>
<th>
    @Html.DisplayNameFor(model => model.nhasanxuat.tennsx)
</th>
<th></th>
</tr>
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.mahang)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.tenhang)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.donvitinh)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.dongia)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.loaihanghoa.tenloai)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.nhasanxuat.tennsx)
        </td>
        <td>
            @Html.ActionLink("Sửa", "suaHH", new { id = item.mahang }) |
            @Html.ActionLink("Chi tiết", "xemHH", new { id = item.mahang }) |
            @Html.ActionLink("Xóa", "xoaHH", new { id = item.mahang })
        </td>
    </tr>
}
</table>
```

f) Kết quả thực thi :

Bài 7, 8: ASP.NET MVC, JQUERY VÀ TRUY XUẤT CSDL BẰNG LINQ

Header

Trang chủ

Trang chủ

Trang chủ

Trang chủ

Đăng nhập

Đăng ký

Quản lý Nhà sản xuất

Quản lý Nhân viên

Quản lý Khách hàng

Quản lý Loại hàng hóa

Quản lý Hàng hóa

Quản lý hàng hóa

[Thêm hàng hóa](#)

Mã hàng hóa	Tên hàng	Đơn vị tính	Đơn giá	Tên loại hàng	Tên nhà sản xuất	
HH0001	CPU i7 Ge4	cái	7000000	cpu	Công ty Intel Việt Nam	Sửa Chi tiết Xóa
HH0002	CPU i5 Ge5	cái	6000000	cpu	Công ty Intel Việt Nam	Sửa Chi tiết Xóa
HH0003	CPU i3 Ge4	cái	5000000	cpu	Công ty Intel Việt Nam	Sửa Chi tiết Xóa
HH0004	RAM 8GB	thanh	2000000	ram	KingMax	Sửa Chi tiết Xóa
HH0005	RAM 4GB	thanh	1000000	ram	KingMax	Sửa Chi tiết Xóa
HH0006	RAM 8GB 16000	thanh	2200000	ram	Công ty Intel Việt Nam	Sửa Chi tiết Xóa


Footer

2/ Bài tập tổng hợp:

- a. SV thực hiện cho mục “Chi tiết”: hiển thị thông tin chi tiết hàng hóa như View sau:

Thông tin hàng hóa

Hình ảnh:



Mã hàng: lgtivi001
Tên hàng: Tivi LED LG 40 inch
Đơn vị tính: cái
Đơn giá: 4500000

Mã loại hàng hóa	Tên loại hàng hóa
tivi	Tivi

Mã nhà sản xuất	Tên nhà sản xuất	Địa chỉ
lg	LG	450 Nguyễn Huệ

[Trở về](#)


Bài 7, 8: ASP.NET MVC, JQUERY VÀ TRUY XUẤT CSDL BẰNG LINQ

- b. SV thực hiện cho mục “Thêm hàng hóa”: dùng để thêm Hàng hóa vào CSDL. Giao diện cho chức năng Thêm hàng hóa như hình :

Thêm hàng hóa

[Trở về](#)

Hình ảnh



Thông tin hàng hóa

Mã hàng

Tên hàng

Đơn vị tính

Đơn giá

Mã loại hàng hóa

Tên loại hàng hóa
Máy giặt

Mã nhà sản xuất

Tên nhà sản xuất
Sony

Địa chỉ
567 Trần Hưng Đạo

Trong đó:

- Mã loại hàng hóa” là DropDownList hiển thị Danh sách Mã loại hàng hóa.
- “Mã nhà sản xuất” là DropDownList hiển thị Danh sách Mã nhà sản xuất.
- “Hình ảnh” là thẻ `<input type="file" ... />` cho phép chọn file chứa hình ảnh đại diện cho Hàng hóa.

Yêu cầu:

- Sử dụng JQuery Ajax, Khi lựa chọn Mã loại hàng hóa trong DropDownList thì hiển thị Thông tin loại hàng hóa tương ứng trên giao diện (hình trên).
- Sử dụng JQuery Ajax, Khi lựa chọn Mã nhà sản xuất trong DropDownList thì hiển thị Thông tin nhà sản xuất tương ứng trên giao diện (hình trên).
- Khi lựa chọn File hình ảnh trong thẻ `<input type="file" ... />` thì hiển thị hình tương ứng trên giao diện (hình trên).
- Nút “Thêm” dùng để thêm Thông tin hàng hóa vào CSDL.
- Kiểm tra ràng buộc về khóa chính.

Bài 7, 8: ASP.NET MVC, JQUERY VÀ TRUY XUẤT CSDL BẰNG LINQ

- c. Mục “Xóa” dùng để xóa Thông tin hàng hóa trong CSDL. Hiện thực ràng buộc về khóa ngoại.
- d. Mục “Sửa” dùng để sửa Thông tin hàng hóa trong CSDL.

IV. NỘI DUNG THỰC HÀNH BÀI 8:

1/ Chương trình mẫu: thực hiện chức năng hiển thị danh sách hàng hóa (trang chủ)

- a) Thêm controller là **HomeController** và nhập code cho HomeController:

```
public class HomeController : Controller
{
    private QLBDHDBContext db = new QLBDHDBContext();
    public ActionResult Index()
    {
        return View(db.hanghoa);
    }
}
```

- b) Thêm view **Index** trong thư mục **Home** và nhập lại cho trang **Index.cshtml**:

```
@model IEnumerable<QLBH.Models.hanghoa>

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/_LayoutPage1.cshtml";
}

<h2>Danh sách hàng hóa</h2>
@{
    int k = 0;
    int socot = 4;

    int sodong = 0;
    int temp = Model.ToList().Count / socot;
    if (Model.ToList().Count % socot == 0)
    {
        sodong = temp;
    }
    else
    {
        sodong = temp + 1;
    }
}
@for (int i = 0; i < sodong; i++)
{
    <div>
        <div class="row">
            @for (int j = 0; j < socot; j++)
```

```
{
    k = i * socot + j;
    if (k >= Model.ToList().Count)
    {
        break;
    }
    var item = Model.ToList()[i * socot + j];

    <div class="col-md-3">
        @using (Html.BeginForm("chonHanghoa", "Home", new { id =
item.mahang })))
        {
            <div>
                <span class="font-weight-bold">Mã hàng: </span>
                @item.mahang
            </div>
            <div>
                Tên hàng: @item.tenhang
            </div>
            <div>
                
            </div>
            <div>
                Đơn giá: @item.dongia
            </div>
            <div>
                <label class="font-weight-bold"
for="txtSoluong">Số lượng: </label>
                <input class="col-6" type="number" value="1"
name="txtSoluong" id="txtSoluong" />
            </div>
            <div>
                <input type="submit" class="btn btn-primary"
value="Chọn mua" />
            </div>
        }
    </div>
}
@if (k >= Model.ToList().Count)
{
    break;
}
</div>
<hr />
</div>
}
```

c) Kết quả thực hiện :

Bài 7, 8: ASP.NET MVC, JQUERY VÀ TRUY XUẤT CSDL BẰNG LINQ

Trang chủ

Trang chủ

Trang chủ

Trang chủ

Đăng nhập

Đăng ký

Quản lý Nhà sản xuất

Quản lý Nhân viên

Quản lý Khách hàng


Quản lý Loại hàng hóa

Quản lý Hàng hóa

Danh sách hàng hóa

Mã hàng: HH0001

Tên hàng: CPU i7 Ge4




Đơn giá: 7000000

Số lượng:

Chọn mua

Mã hàng: HH0002

Tên hàng: CPU i5 Ge5




Đơn giá: 6000000

Số lượng:

Chọn mua

Mã hàng: HH0003

Tên hàng: CPU i3 Ge4




Đơn giá: 5000000

Số lượng:

Chọn mua

Mã hàng: HH0004

Tên hàng: RAM 8GB




Đơn giá: 2000000

Số lượng:

Chọn mua

Mã hàng: HH0005

Tên hàng: RAM 4GB




Đơn giá: 1000000

Số lượng:

Chọn mua

Mã hàng: HH0006

Tên hàng: RAM 8GB 16000




Đơn giá: 2200000

Số lượng:

Chọn mua

Mã hàng: lgmaygiat001

Tên hàng: Máy giặt LG 6kg cửa trên




Đơn giá: 39000000

Số lượng:

Chọn mua

Mã hàng: lgtivi001

Tên hàng: Tivi LG LED 40 inch



Đơn giá: 4500000

Số lượng:

Chọn mua

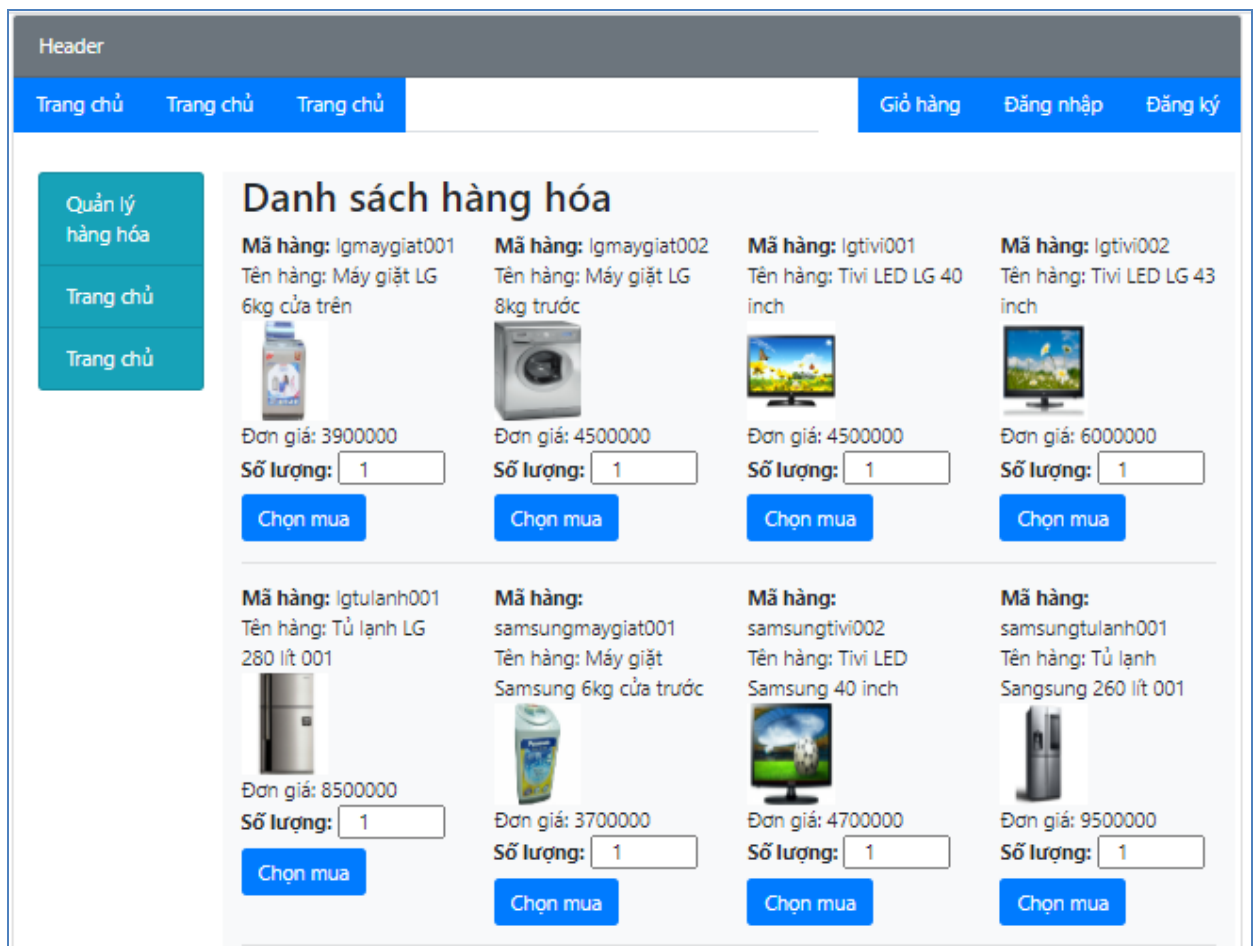
2/ Bài tập tổng hợp:

- Hiện thực chức năng “Lập giỏ hàng”. Khi Khách hàng nhấn nút “Chọn mua” trong view ở Hình 8.1 thì Hàng hóa tương ứng được thêm vào Giỏ hàng như Hình 8.2a hay Hình 8.2b.
- Mục “Giỏ hàng” dùng để hiển thị view “Giỏ hàng” như Hình 8.2a hay Hình 8.2b hay Hình 8.2c.
- Mục “Trở về” trong view ở hình 8.2(a,b,c) hiển thị view như Hình 8.1.
- Mục “Xóa” trong view ở Hình 8.2(a,b) dùng để xóa hàng hóa tương ứng trong Giỏ hàng.
- Nút “Lập đơn đặt hàng” trong view ở Hình 8.2b dùng để lập phiếu đặt hàng và lưu vào CSDL.

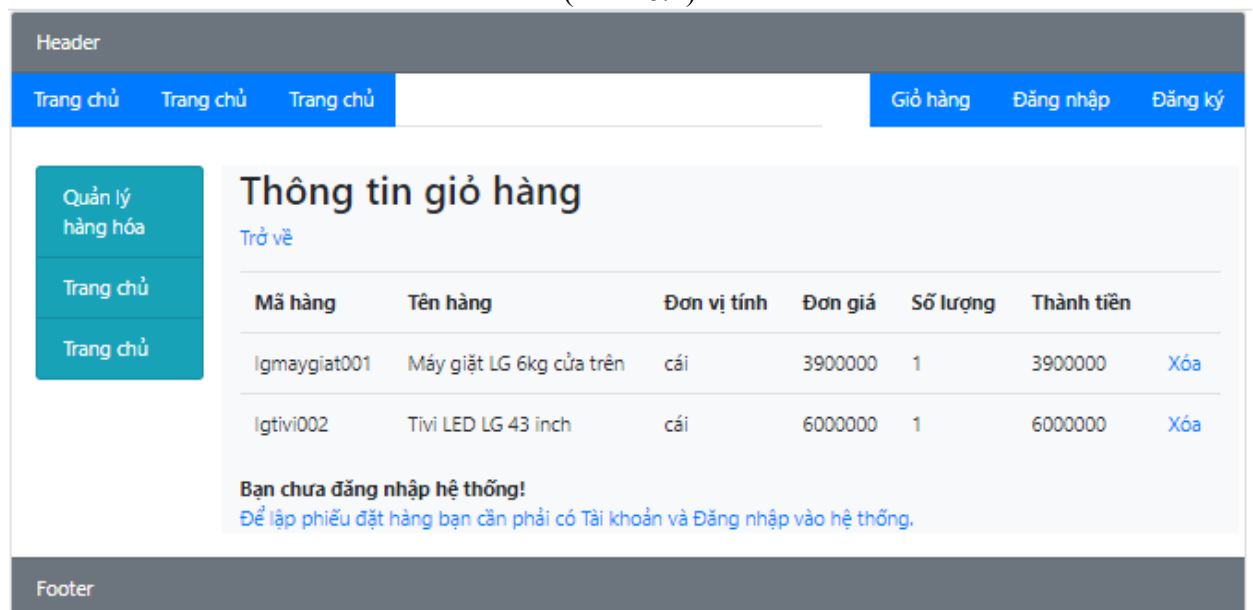
Khoa CNTT – Trường ĐH Công Nghệ Sài Gòn

Trang 76

Bài 7, 8: ASP.NET MVC, JQUERY VÀ TRUY XUẤT CSDL BẰNG LINQ



(Hình 8.1)



(Hình 8.2a: Khách hàng chưa đăng nhập hệ thống)

Bài 7, 8: ASP.NET MVC, JQUERY VÀ TRUY XUẤT CSDL BẰNG LINQ

Header

Trang chủTrang chủTrang chủ

Giỏ hàngĐăng xuấtPhúc

Quản lý hàng hóa

Trang chủ

Trang chủ

Thông tin giỏ hàng

[Trở về](#)

Mã hàng	Tên hàng	Đơn vị tính	Đơn giá	Số lượng	Thành tiền	
Igmaygiat001	Máy giặt LG 6kg cửa trên	cái	3900000	1	3900000	Xóa
Igtivi001	Tivi LED LG 40 inch	cái	4500000	1	4500000	Xóa

Phiếu đặt hàng

Mã phiếu đặt hàng

Ngày đặt hàng

Ngày giao hàng

Địa chỉ giao hàng

Lập đơn đặt hàng

Footer

(Hình 8.2b: Khách hàng đã đăng nhập vào hệ thống)

Header

Trang chủTrang chủTrang chủ

Giỏ hàngĐăng xuấtPhúc

Quản lý hàng hóa

Trang chủ

Trang chủ

Thông tin giỏ hàng

[Trở về](#)

Mã hàng	Tên hàng	Đơn vị tính	Đơn giá	Số lượng	Thành tiền
Bạn chưa có Hàng hóa trong Giỏ hàng!					
Bạn cần chọn Hàng hóa cần mua và thêm vào Giỏ hàng.					

Footer

(Hình 8.2c: Khách hàng đã đăng nhập vào hệ thống nhưng giỏ hàng chưa có)

