



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
VNUHCM - UIT

LẬP TRÌNH WEB

CHƯƠNG 6 TRUY CẬP CƠ SỞ DỮ LIỆU

NỘI DUNG

- 1 • Tạo CSDL trước (Database First)
- 2 • Chuỗi kết nối CSDL
- 3 • Chú thích dữ liệu (Data Annotation)
- 4 • Truy cập mô hình dữ liệu từ Controller
- 5 • Tùy chỉnh trang index (index.cshtml)
- 6 • Phân trang trong ASP.NET MVC

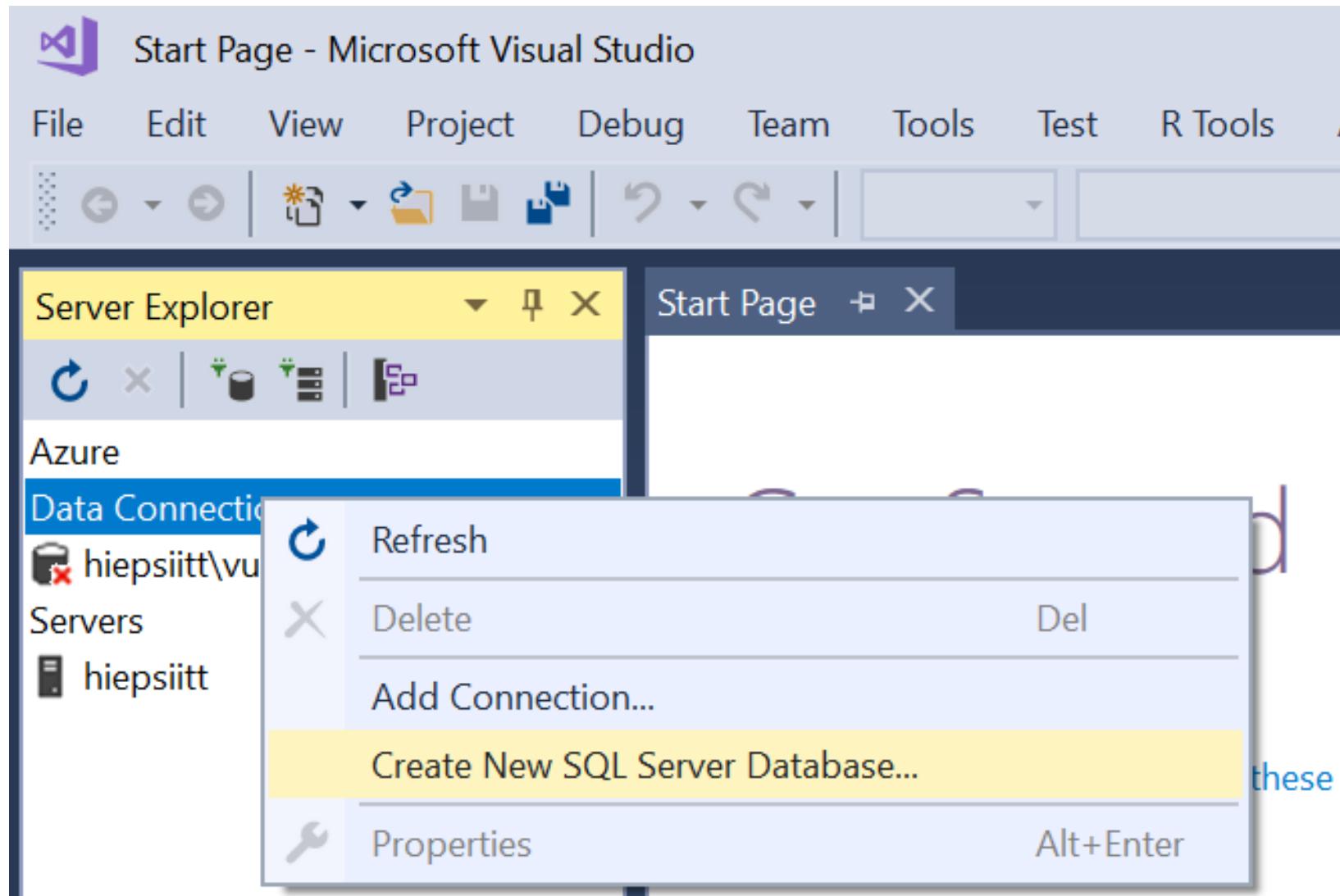
1. TẠO CSDL TRƯỚC (DATABASE FIRST)

- ❖ Trong phần này chúng ta xây dựng 1 ứng dụng ASP.NET MVC dùng mô hình Database First.
- ❖ Database First cho phép chúng ta đảo ngược 1 mô hình từ 1 database đã tồn tại từ trước. Mô hình này được lưu trong tập tin tên là **EDMX (.edmx)** và có thể xem và chỉnh sửa trong Entity Framework Designer (phần thiết kế Entity Framework). Các lớp tương tác trong ứng dụng sẽ được tự động phát sinh từ tập tin EDMX.

TẠO 1 DATABASE

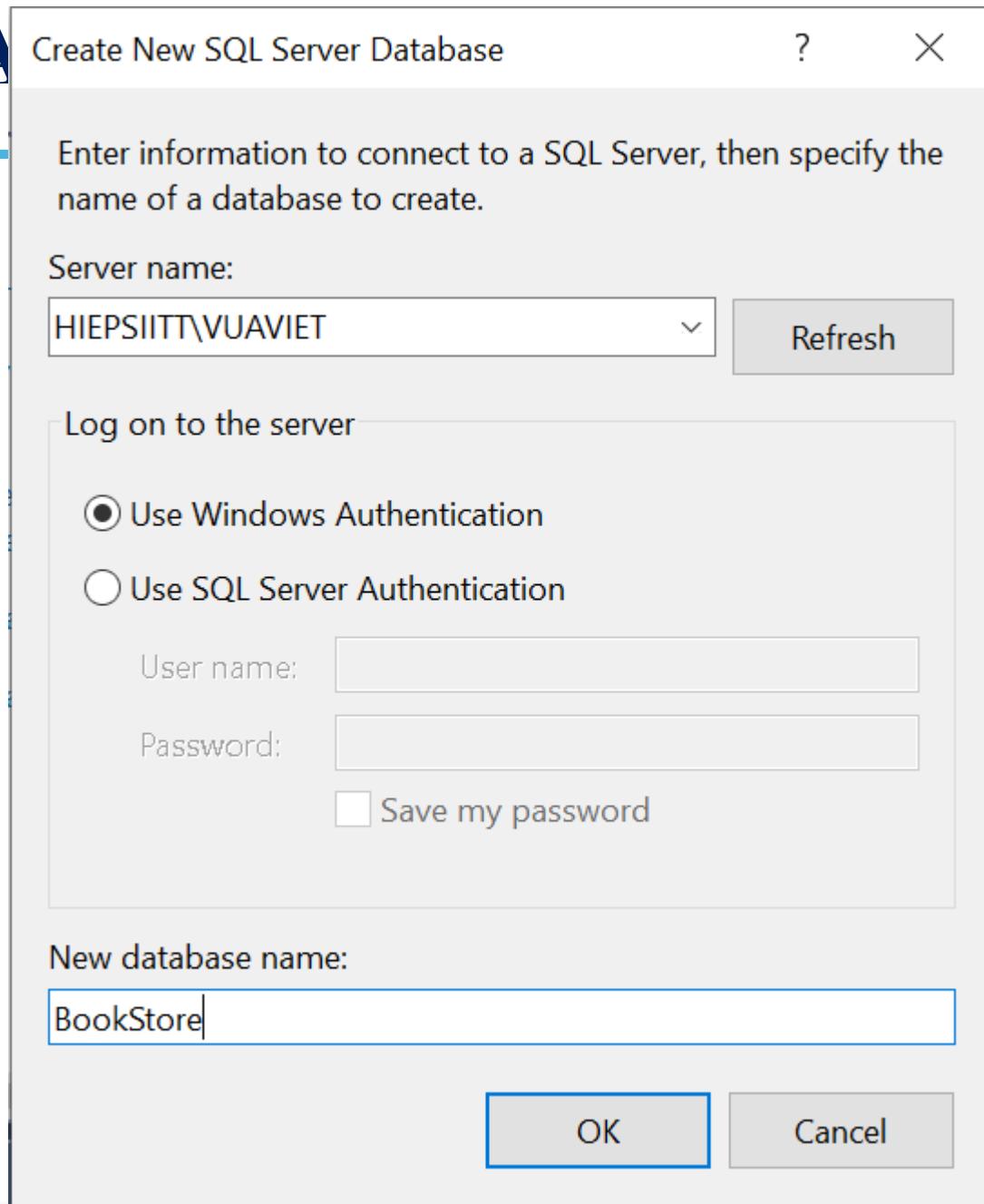
- ❖ Trước hết, bạn phải tạo 1 database để có thể truy cập dữ liệu. Server database có thể cài đặt ở những nơi khác nhau tùy thuộc vào phiên bản Visual Studio.
- ❖ Để tạo database trong Visual Studio, chọn View -> Server Explorer. Click chuột phải vào Database Connections -> Add Connections... Bạn có thể kết nối với (localdb)\v11.0 hoặc SQL Express .\SQLEXPRESS tùy theo cái nào bạn đã cài. Sau đó điền tên database là **BookStore**.

TẠO 1 DATABASE



TẠO 1 DATABASE

- ❖ Tiếp theo
Chọn Server
name → Tạo
CSDL



TẠO 1 DATABASE

- ❖ Tiếp đến chọn OK, bạn sẽ được hỏi tạo mới database vừa đặt tên, chọn Yes để tạo mới database.
- ❖ Sau đó, chuột phải lên database vừa tạo, chọn New Query và copy & paste đoạn script sql, và nhấn Excute (Ctrl + Shift + E) để thực thi.

TẠO 1 DATABASE

```
create table Category(
    CategoryID int identity(1,1) primary key,
    CategoryName nvarchar(100) not null
)

CREATE TABLE Author(
    AuthorID int identity(1,1) primary key,
    AuthorName nvarchar(50) NOT NULL,
    AuthorEmail nvarchar(50) NULL,
    AuthorAddress nvarchar(150) NULL
)

create table Book(
    BookID int identity(1,1) primary key,
    Title nvarchar(200) not null,
    AuthorID int not null,
    Price decimal(18,0),
    Images varchar(200),
    CategoryID int not null,
    Description ntext,
    Published datetime default(getdate()),
    ViewCount int default(0),
    constraint fk_book_category foreign key (CategoryID) references Category(CategoryID),
    constraint fk_book_author foreign key(AuthorID) references Author(AuthorID)
)
```

TẠO 1 DATABASE

The screenshot shows the SSMS interface with the following details:

- Server Explorer:** Shows a connection to "hiepsiitt\vuaviet.BookStore". Under "Tables", there are entries for Author, Book, and Category.
- SQLQuery1.sql:** The current query window contains T-SQL code to create a "Book" table. The code includes columns for BookID (primary key, identity), Title, AuthorID, Price, Images, CategoryID, Description, Published (default getdate()), ViewCount (default 0), and two foreign key constraints (fk_book_category and fk_book_author).
- Status Bar:** Shows "82 %", "T-SQL", "Message", and "Command(s) completed successfully."

```
12 go
13 create table Book(
14     BookID int identity(1,1) primary key,
15     Title nvarchar(200) not null,
16     AuthorID int not null,
17     Price decimal(18,0),
18     Images varchar(200),
19     CategoryID int not null,
20     Description ntext,
21     Published datetime default(getdate()),
22     ViewCount int default(0),
23     constraint fk_book_category foreign key (CategoryID) references Category(CategoryID),
24     constraint fk_book_author foreign key(AuthorID) references Author(AuthorID)
25 )
26
27
```

TẠO 1 DATABASE

- ❖ Run đoạn script sql để tạo mới các bảng cho database.
- ❖ Chèn dữ liệu mẫu cho database chạy đoạn script (từ file **DataBookStore.txt**):

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Server Explorer pane displays a connection to 'hiepsiitt\vuaviet.BookStore' with expanded sections for Data Connections, Tables (Author, Book, Category), Views, Stored Procedures, Functions, Synonyms, Types, and Assemblies. The 'Book' node under Tables is currently selected. On the right, the main window has three tabs at the top: 'dbo.Book [Data]', 'dbo.Category [Data]', and 'SQLQuery2.sql *'. The 'SQLQuery1.sql' tab is active, showing a T-SQL script for inserting data into the 'Category' table. The script consists of 15 numbered lines:

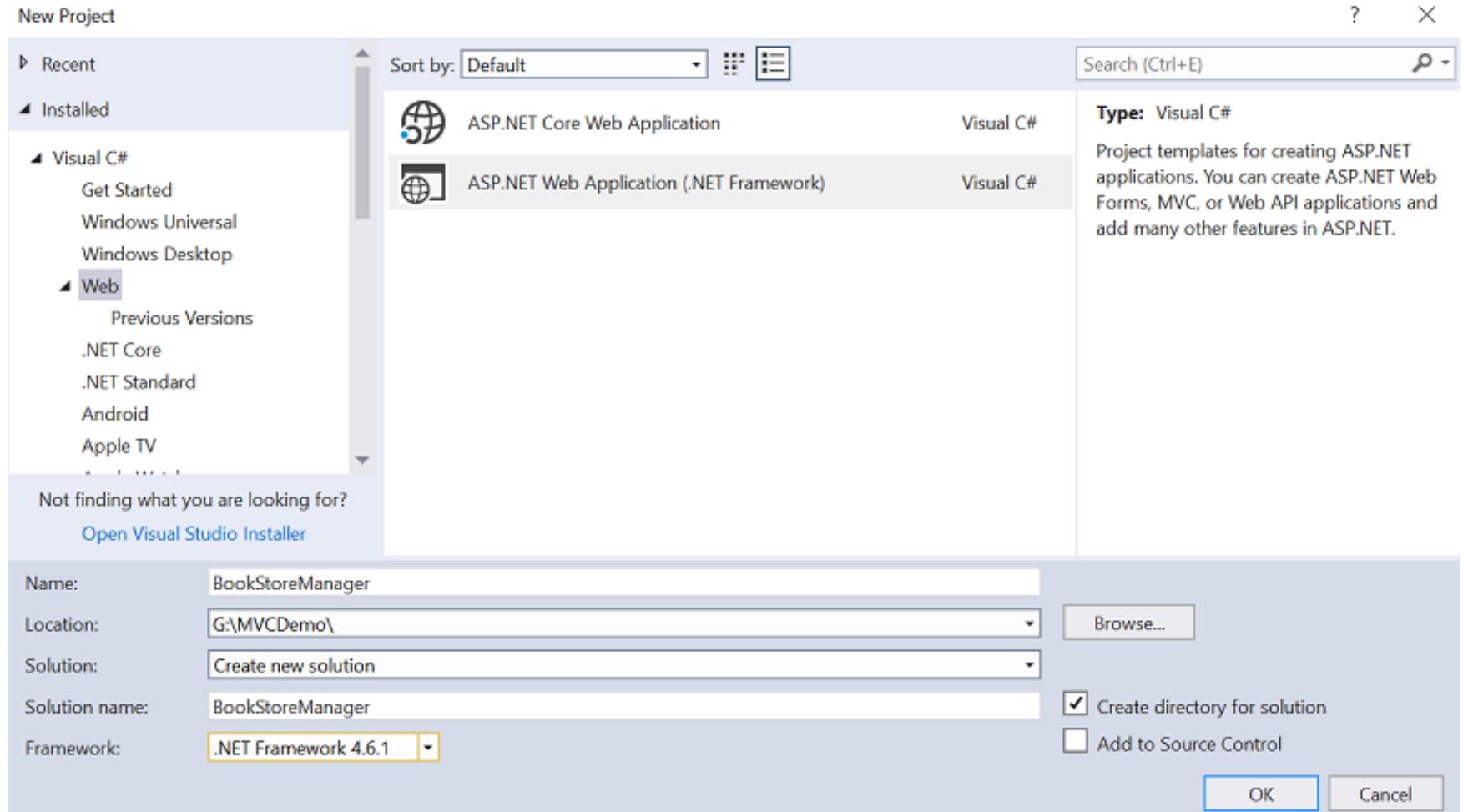
```
1 SET IDENTITY_INSERT [dbo].[Category] ON
2 GO
3 INSERT Category (CategoryID, CategoryName) VALUES (1, N'Ngoại ngữ ')
4 GO
5 INSERT Category (CategoryID, CategoryName) VALUES (2, N'Công nghệ thông tin')
6 GO
7 INSERT Category (CategoryID, CategoryName) VALUES (3, N'Văn học')
8 GO
9 INSERT Category (CategoryID, CategoryName) VALUES (4, N'Khoa học kỹ thuật')
10 GO
11 INSERT Category (CategoryID, CategoryName) VALUES (5, N'Nông nghiệp')
12 GO
13 INSERT Category (CategoryID, CategoryName) VALUES (6, N'Triết học-Chính trị')
14 GO
15 SET IDENTITY_INSERT [dbo].[Category] OFF
```

The status bar at the bottom indicates '82 %' completion. Below the tabs, there are buttons for 'T-SQL', 'Results', and 'Message'.

TẠO ỨNG DỤNG WEB

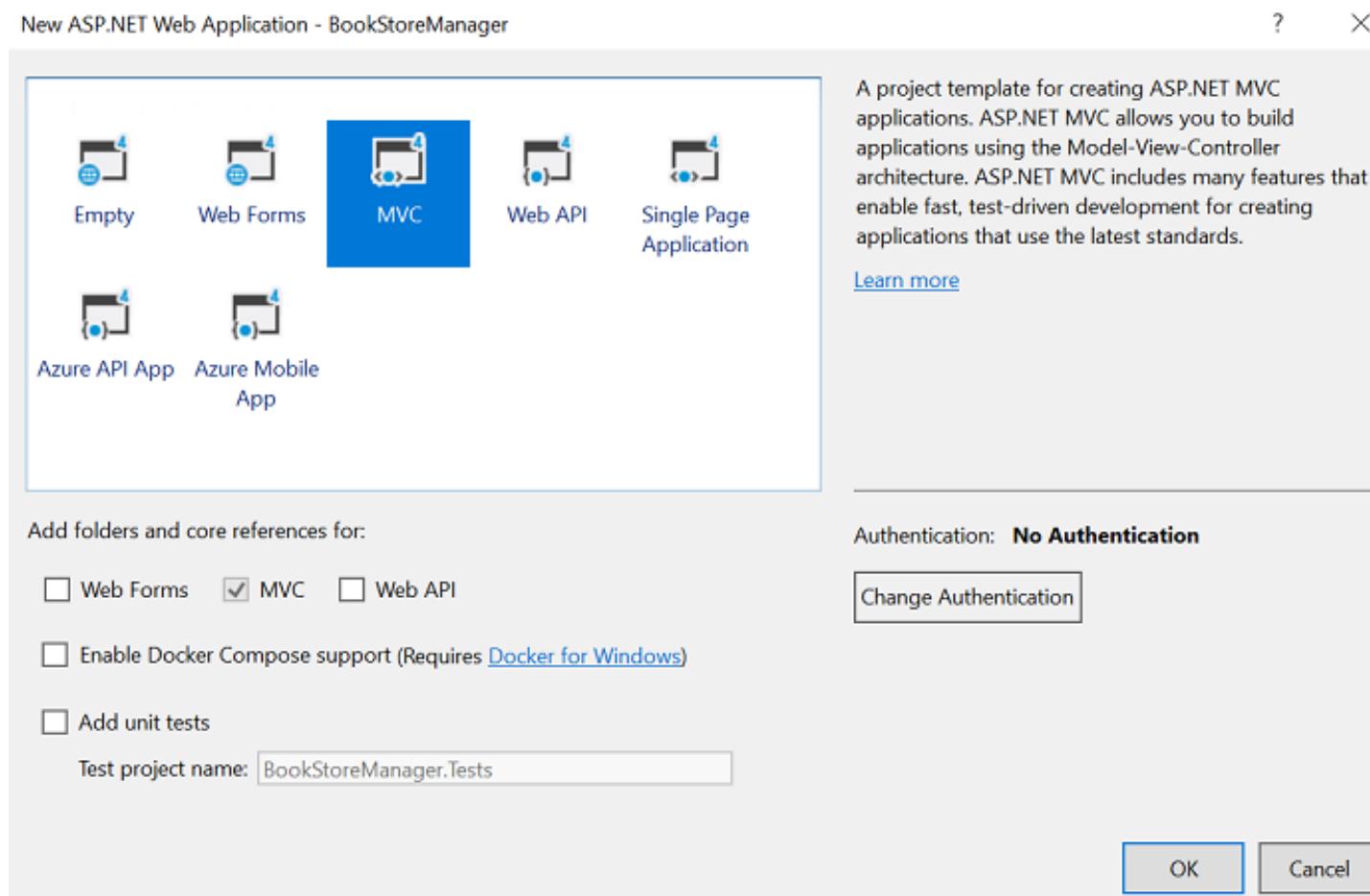
- ❖ Sau khi tạo database xong, chúng ta sẽ xây dựng 1 ứng dụng **Web application** dùng **Database First** để thực hiện truy cập.
- ❖ **Bước 1:** Mở Visual Studio, chọn File → New → Project... Chọn **Visual C#** từ bên menu trái và chọn **ASP.NET Web Application**, đặt tên dự án là **BookStoreManager**, chọn OK.

TẠO ỨNG DỤNG WEB



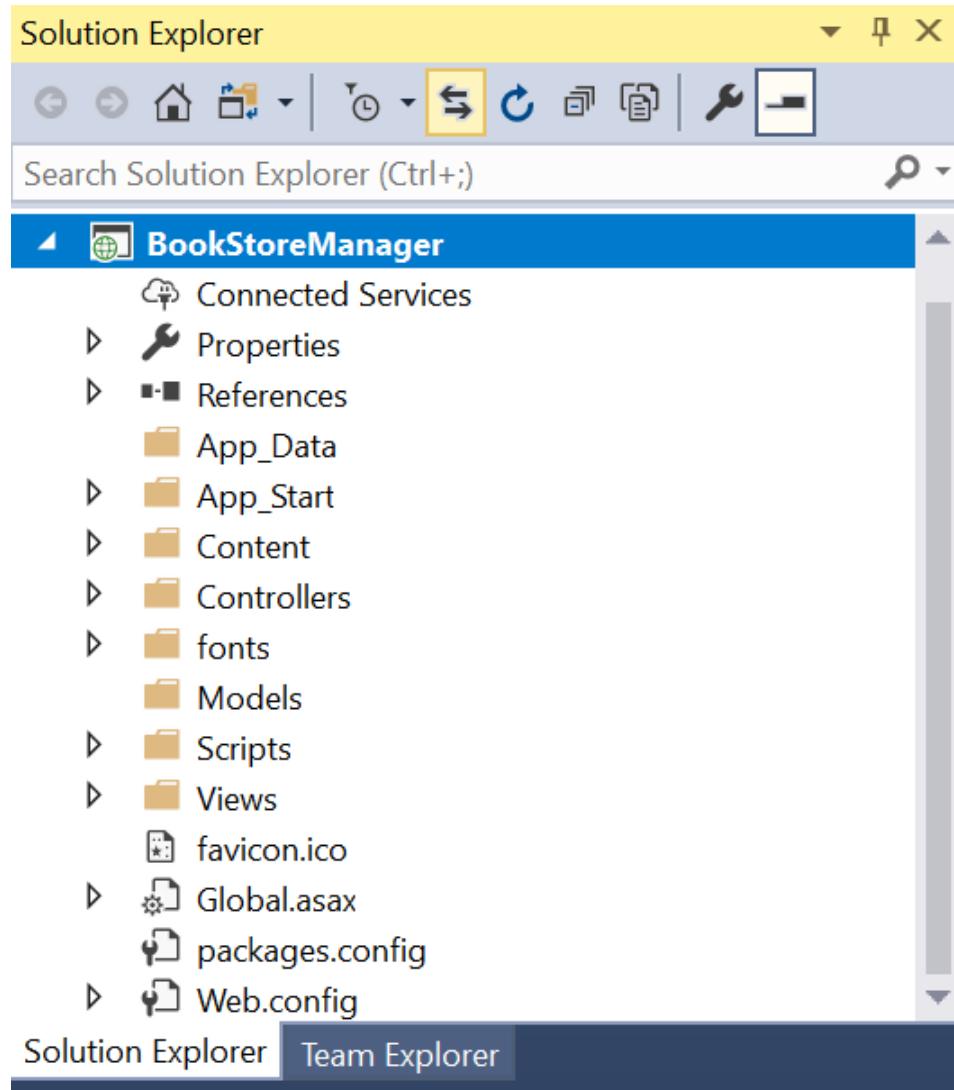
TẠO ỨNG DỤNG WEB

❖ **Bước 2:** Chọn **MVC** sau đó chọn **OK** như hình sau:



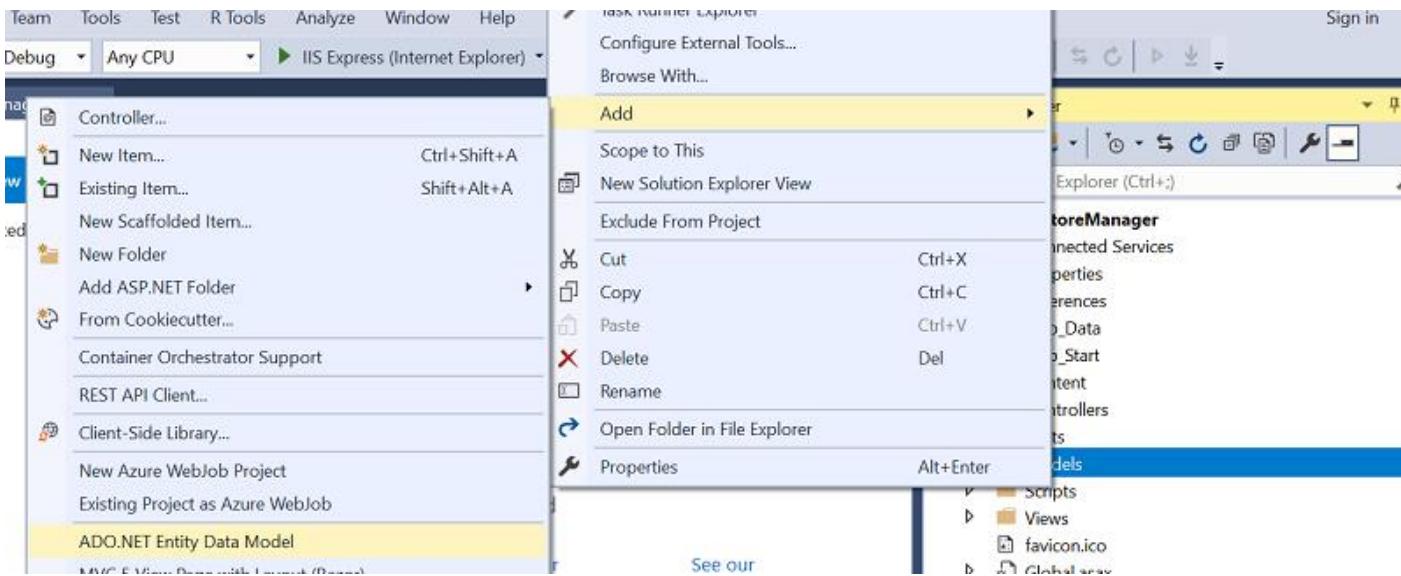
TẠO ỨNG DỤNG WEB

- ❖ Cuối cùng chúng ta
được một ứng dụng
Web:



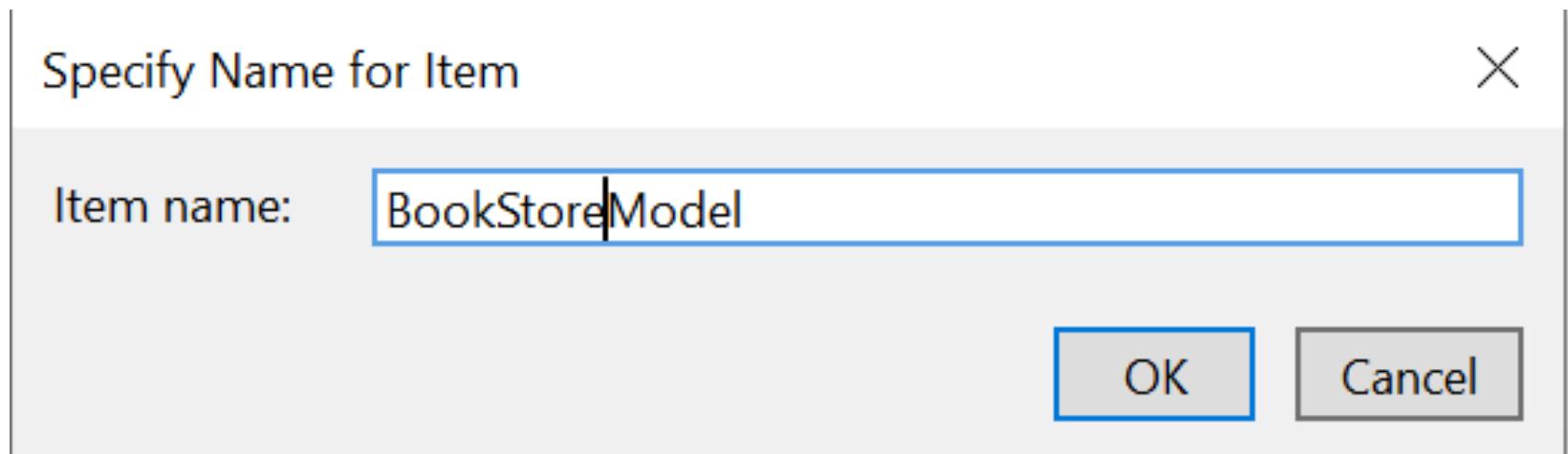
XÂY DỰNG MÔ HÌNH

- ❖ Tiếp theo xây dựng mô hình sử dụng Entity Framework Designer.
- ❖ **Bước 1:** Chọn Project → Models → Add → ADO.NET Entity Data Model.



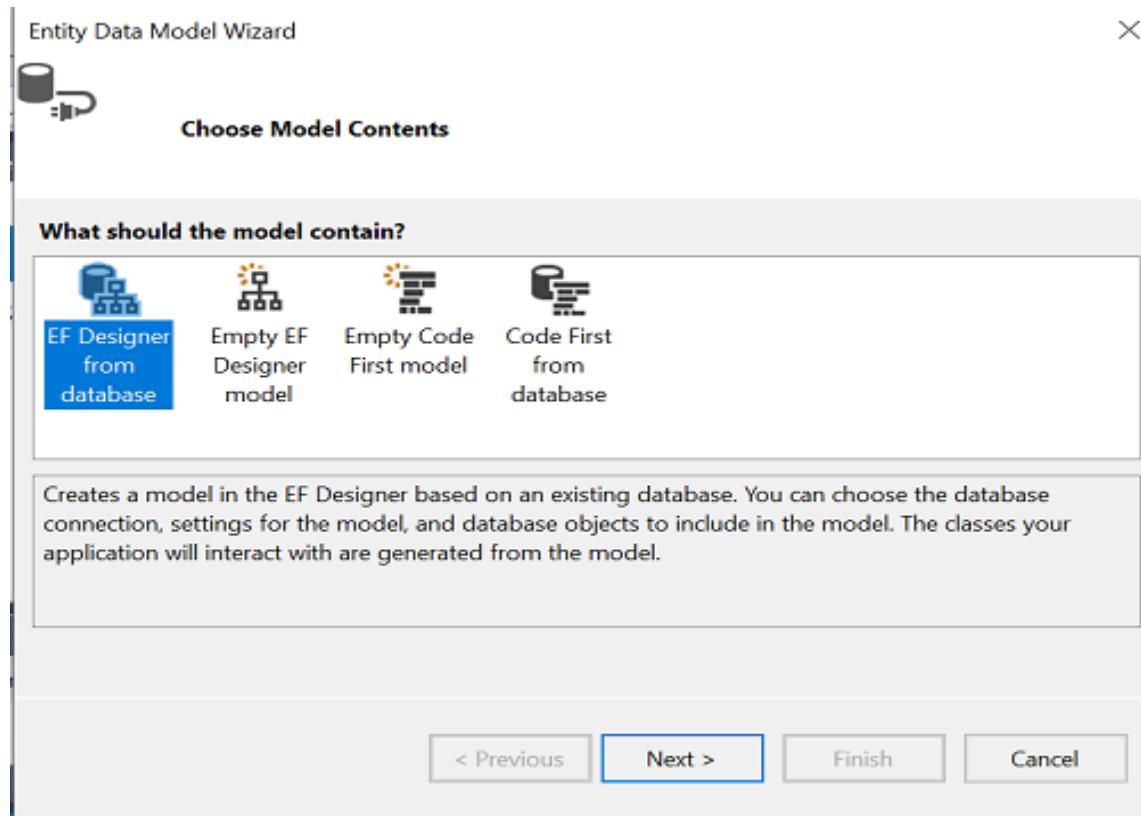
XÂY DỰNG MÔ HÌNH

- ❖ **Bước 2:** điền tên cho mô hình là **BookStoreModel** và nhấn **OK**.



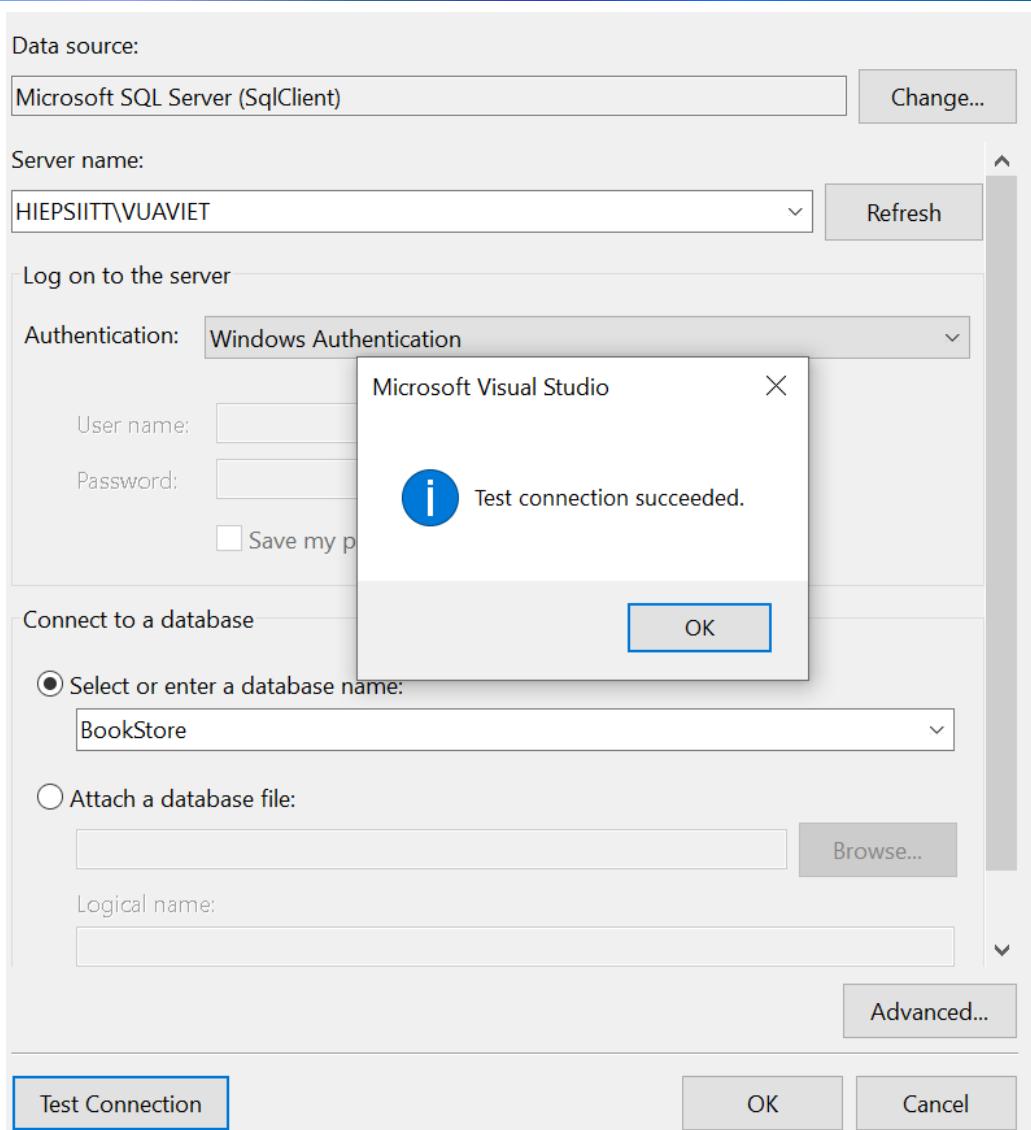
XÂY DỰNG MÔ HÌNH

❖ **Bước 3:** Từ cửa sổ Entity Data Model Wizard sẽ hiện lên, chúng ta chọn EF Designer from database, nhấn Next.



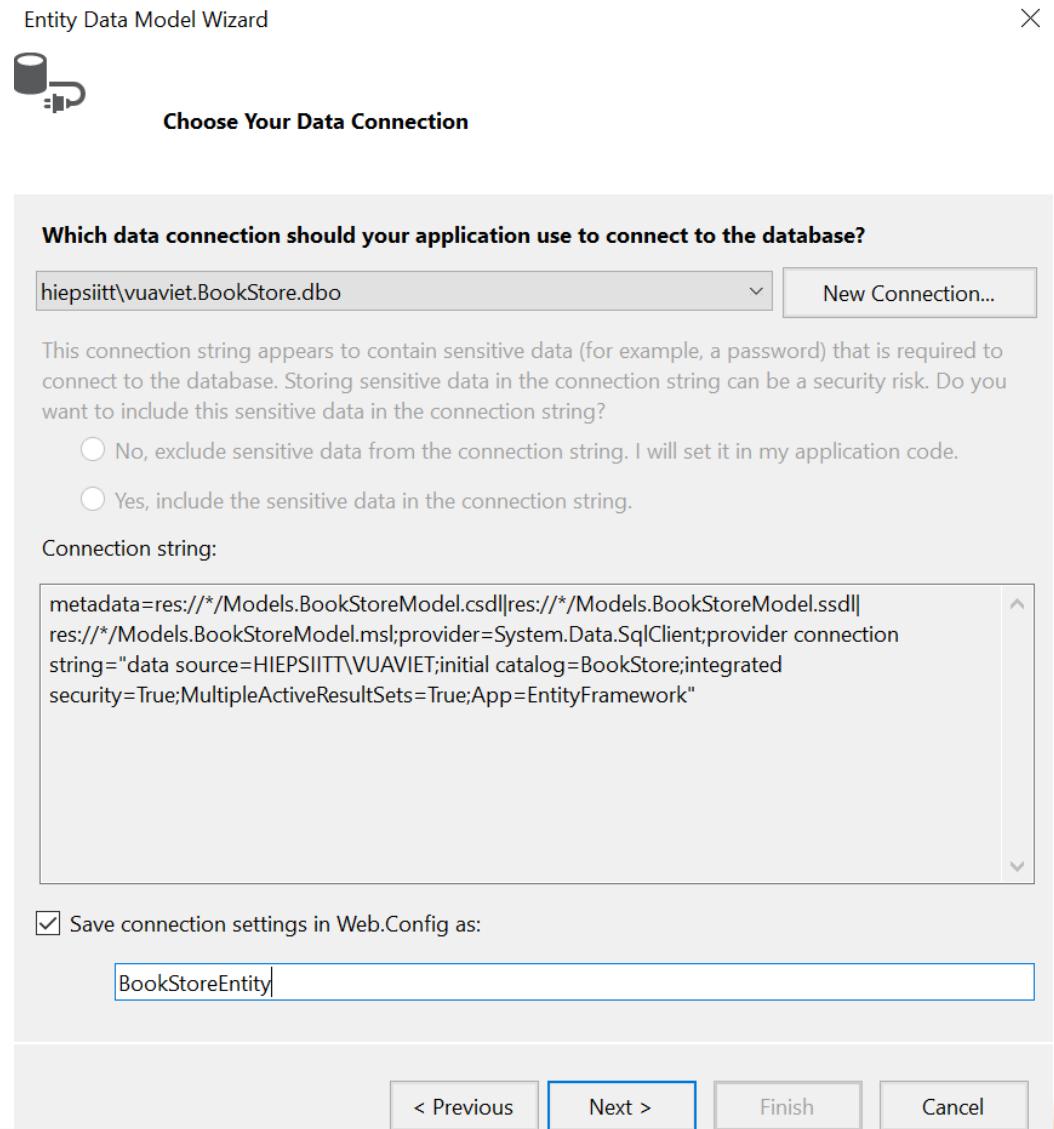
XÂY DỰNG MÔ HÌNH

❖ **Bước 4:** Click vào
New Connection
chọn Server name
→Database
→Click Test
Connection:



XÂY DỰNG MÔ HÌNH

❖ **Bước 5:** đặt tên
cho chuỗi kết nối
tới database là
BookStoreManager
→ Click vào Next



XÂY DỰNG MÔ HÌNH

❖ Bước 6: Chọn Entity Framework 6.X -> Click Next

Entity Data Model Wizard



Choose Your Version

Which version of Entity Framework do you want to use?

- Entity Framework 6.x
- Entity Framework 5.0



It is also possible to install and use other versions of Entity Framework.
[Learn more about this](#)

< Previous

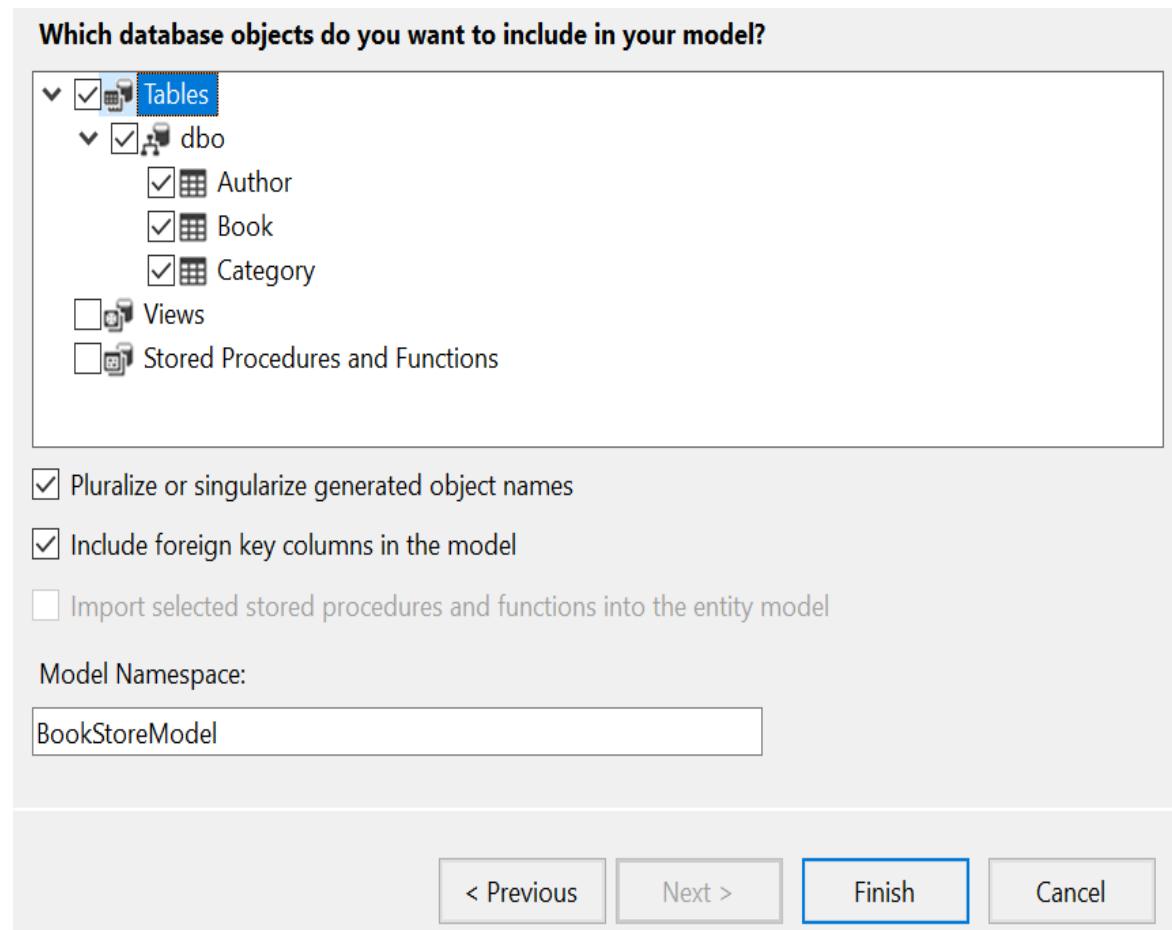
Next >

Finish

Cancel

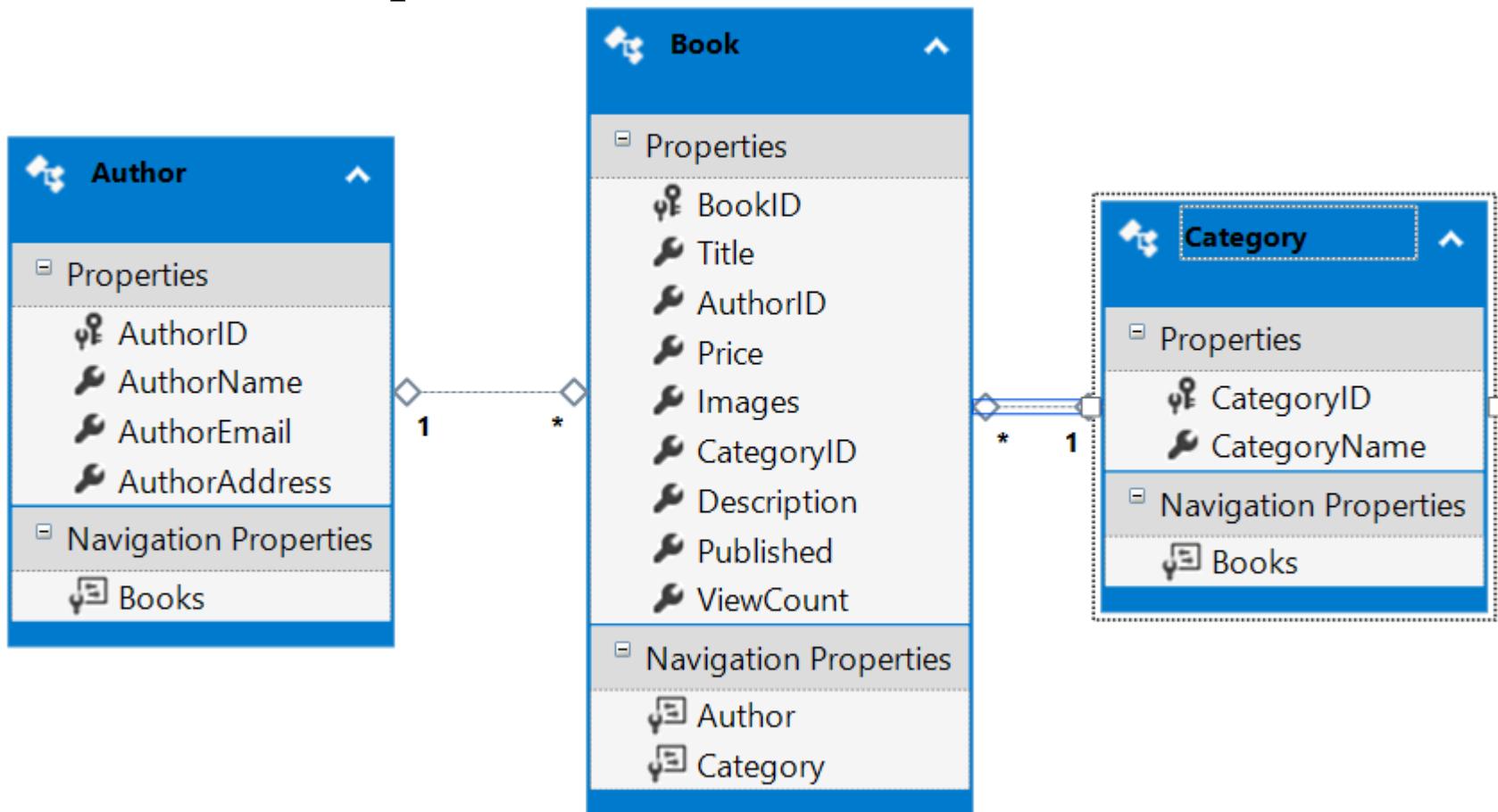
XÂY DỰNG MÔ HÌNH

❖ **Bước 7:** Từ cửa sổ Entity Data Model Wizard, chọn **Tables** để chọn các bảng sẽ được tạo thành các lớp mã nguồn.

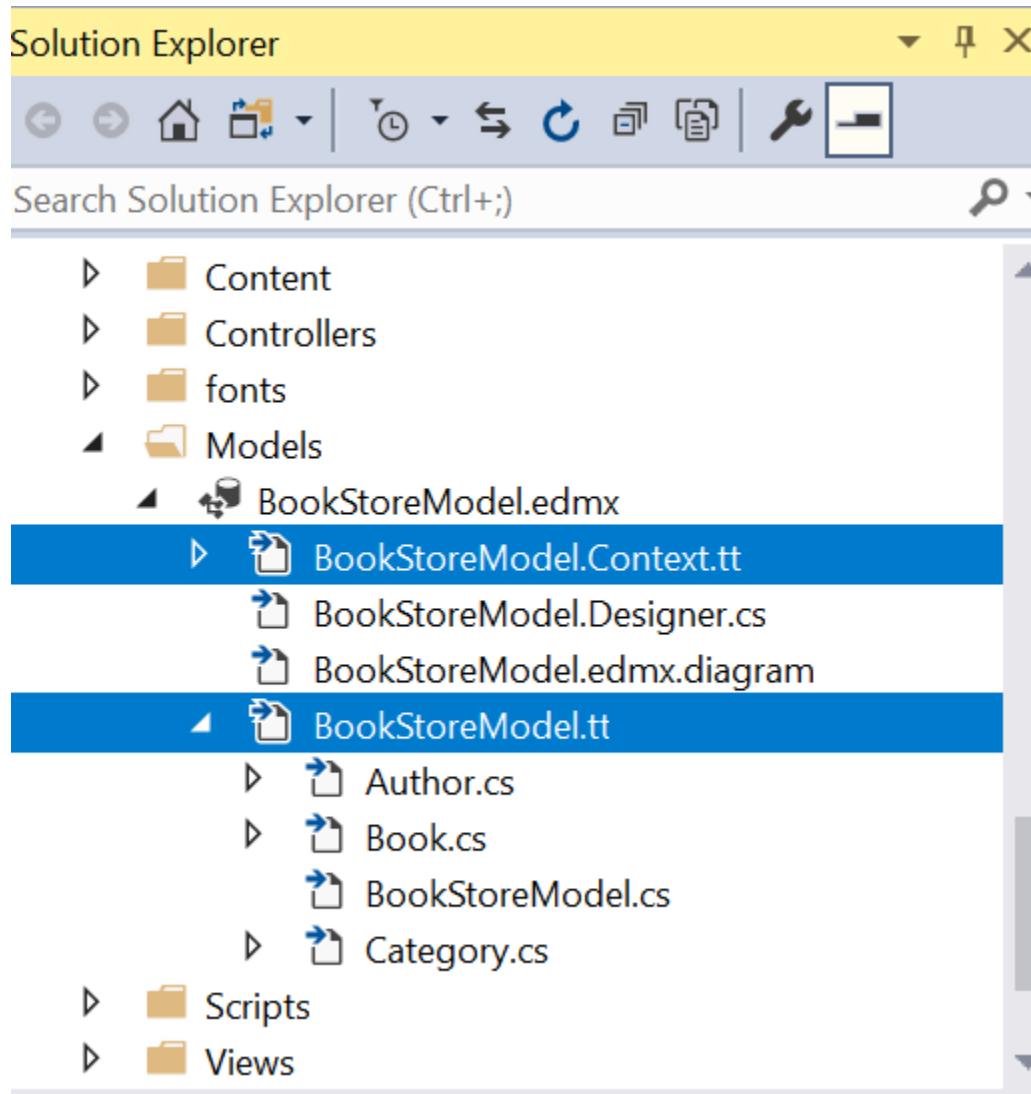


XÂY DỰNG MÔ HÌNH

❖ **Bước 8:** Kết quả mô hình database như hình sau.



ĐỌC VÀ GHI DỮ LIỆU



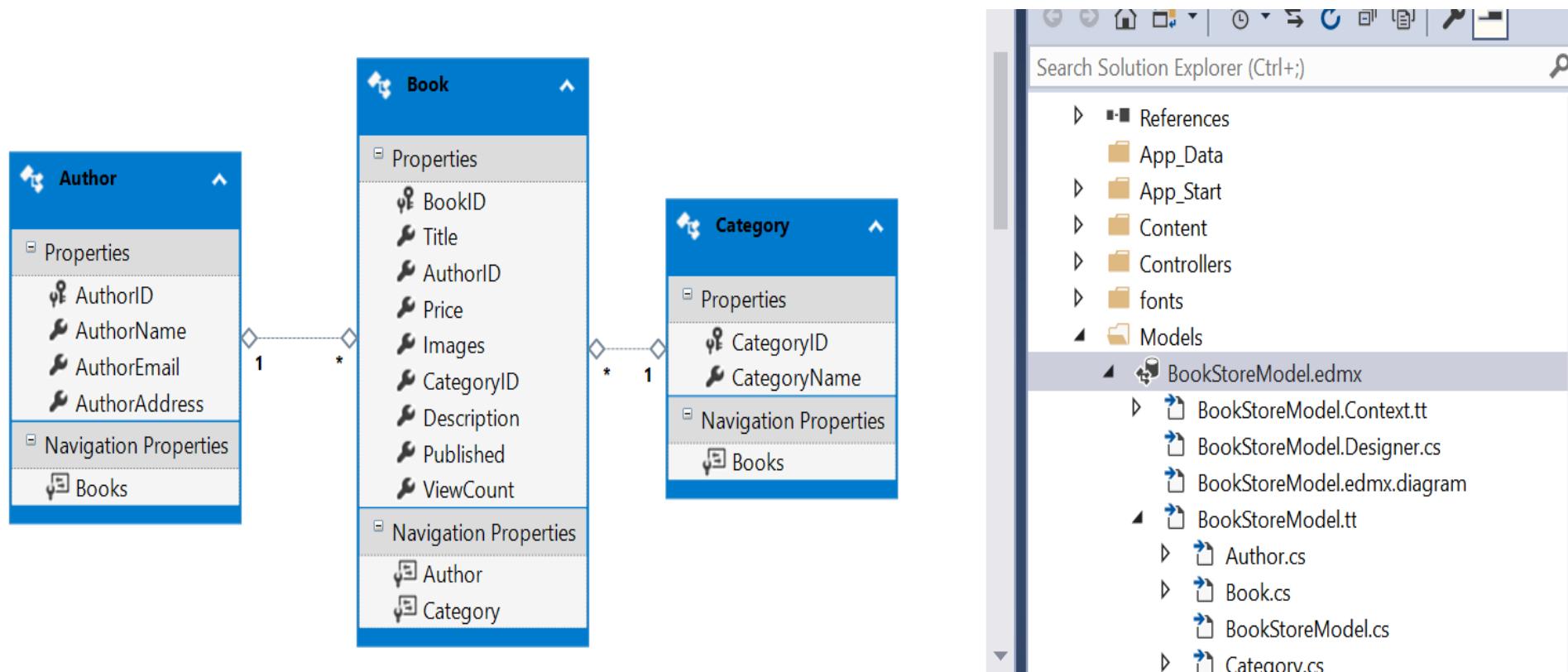
VẤN ĐỀ THAY ĐỔI MÔ HÌNH

- ❖ Nếu database thay đổi thì bạn có thể dễ dàng thay đổi mô hình mã nguồn. Trong database, chúng ta thêm một bảng Users mới bằng đoạn query như sau:

```
create table Users(  
    UserID int identity(1,1) primary key,  
    Username varchar(50) not null,  
    Password nvarchar(50) not null,  
    Fullname nvarchar(100) null,  
    IsAdmin bit default(0)  
)
```

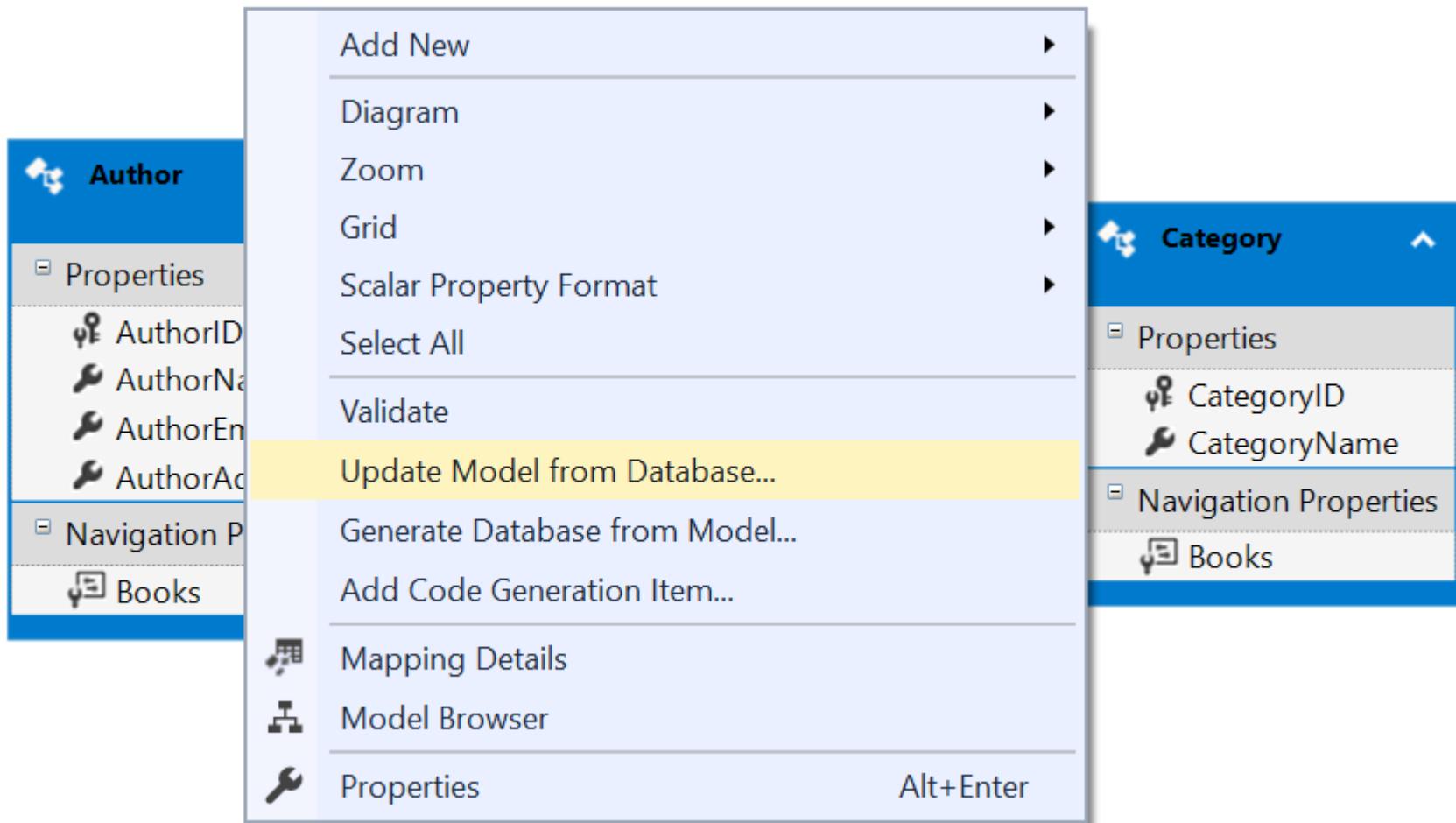
VĂN ĐỀ THAY ĐỔI MÔ HÌNH

- ❖ **Bước 1:** Để cập nhật mô hình, chúng ta mở file BookStoreModel.edmx, như hình sau:



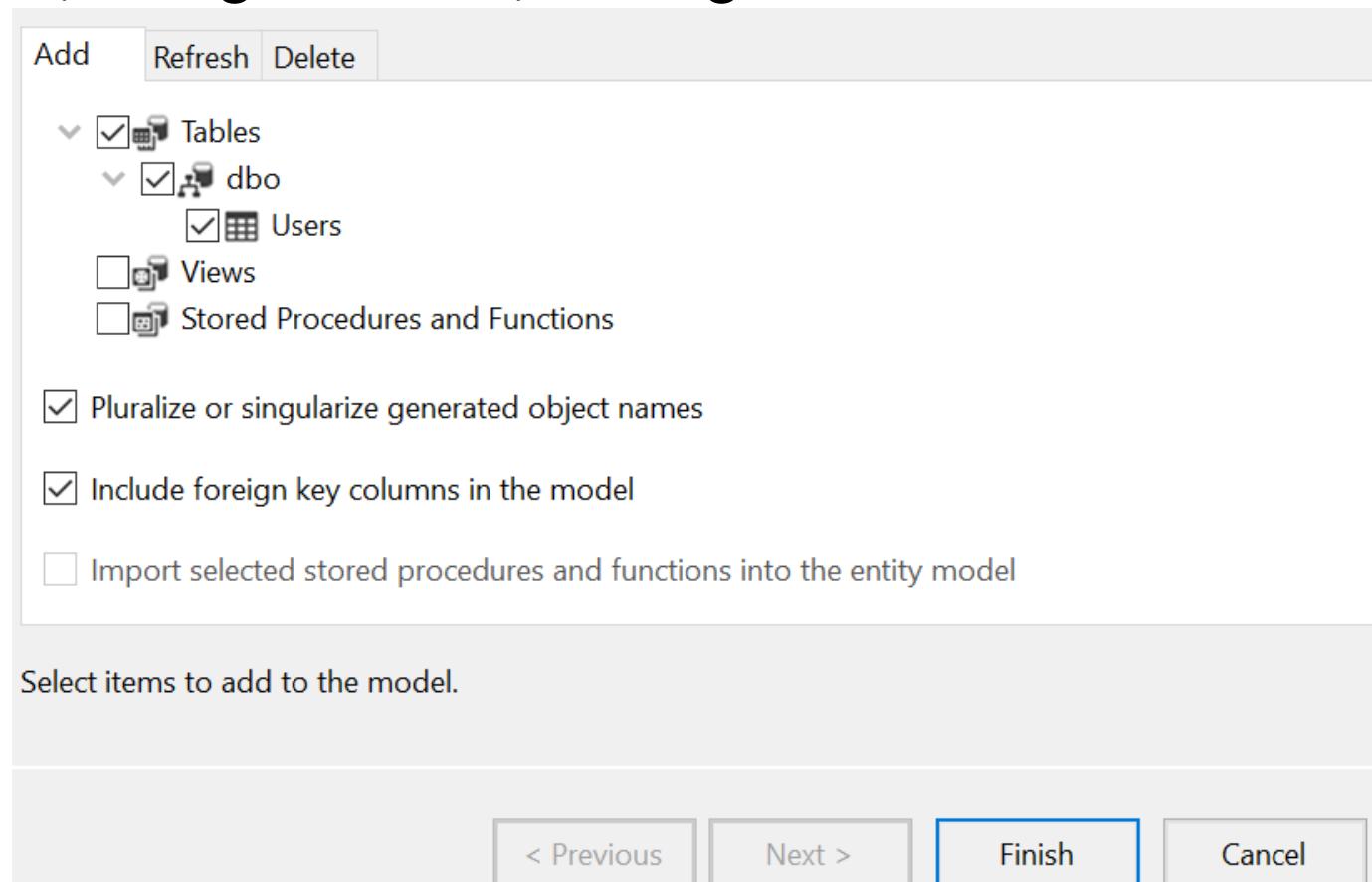
VĂN ĐỀ THAY ĐỔI MÔ HÌNH

❖ **Bước 2:** Click chuột phải chọn **Update Model from**

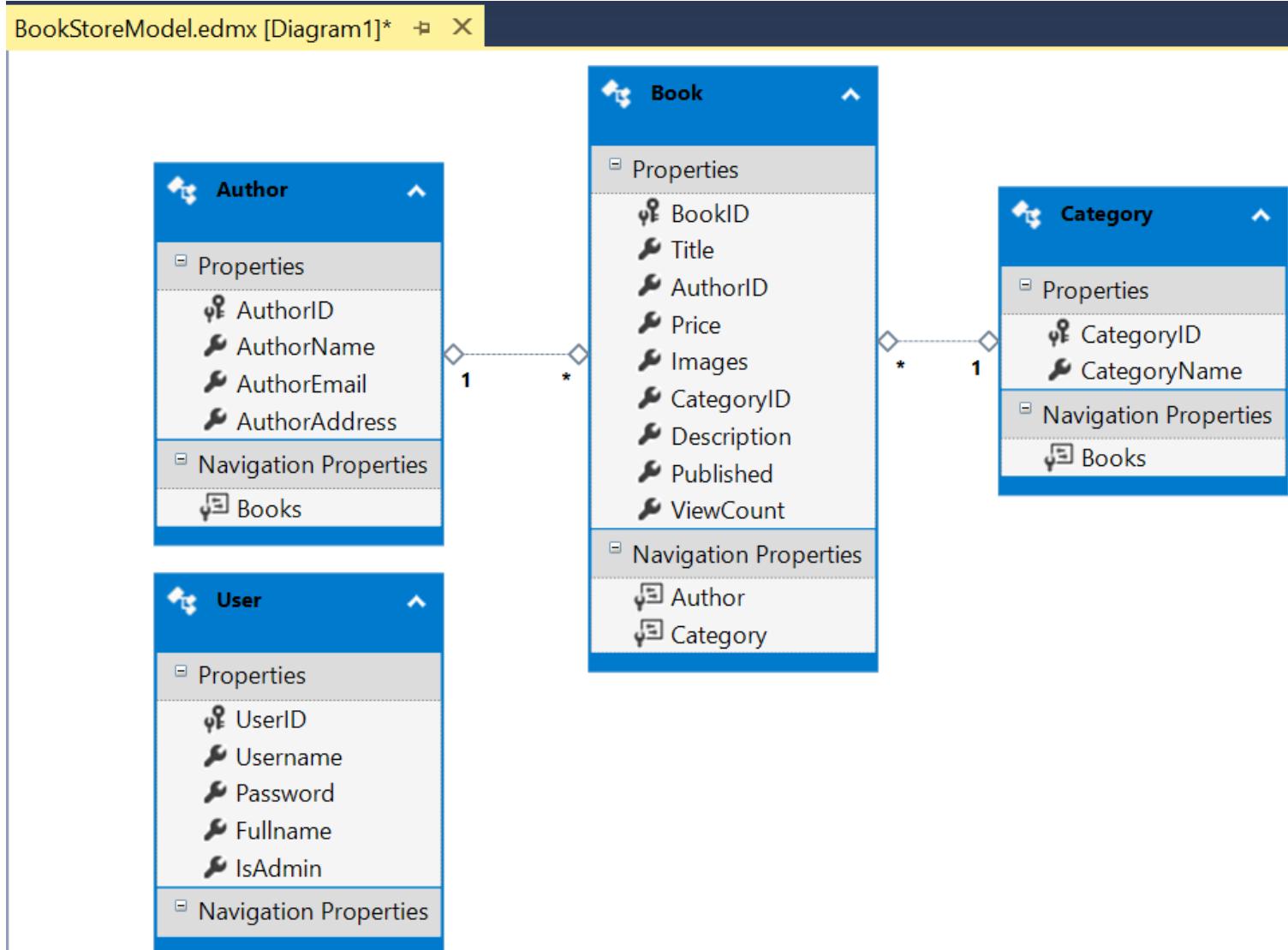


VẤN ĐỀ THAY ĐỔI MÔ HÌNH

- ❖ **Bước 3:** Ở tab Add của Update Wizard chọn Tables, sau đó chọn bảng mới cần tạo mã nguồn.



VẤN ĐỀ THAY ĐỔI MÔ HÌNH



2. CHUỖI KẾT NỐI CSDL (Connection strings)

- ❖ Sau khi tạo được mô hình (Model) ở phần trước, bạn cần tìm hiểu **chuỗi kết nối database** trong ASP.NET MVC.
- ❖ Để xem nội dung chuỗi kết nối, bạn có thể mở tập tin **Web.config** trong dự án và tìm chõ có thẻ **connectionStrings**.

```
<connectionStrings>
    <add name="BookStoreManager" connectionString="metadata=res://*/Models.BookStoreMo
//*/Models.BookStoreModel.ssdl|res://*/Models.BookStoreModel.msl;
provider=System.Data.SqlClient;provider connection string="data source=HIEPSIITT\
</connectionStrings>
```

2. CHUỖI KẾT NỐI CSDL (Connection strings)

- ❖ Ba tập tin `Models.Model.csdl`, `Models.Model.ssdl` và `Models.Model.msl` là các tập tin cấu hình MVC.
- ❖ Server SQL cài mặc định của máy đang dùng (`data source=HIEPSIITT\VUAVIET`) và `tên database` là `initial catalog=BookStore` với chế độ tích hợp bảo mật (`integrated security=True`), không cần tài khoản đăng nhập.

3. CHÚ THÍCH DỮ LIỆU (Data Annotation)

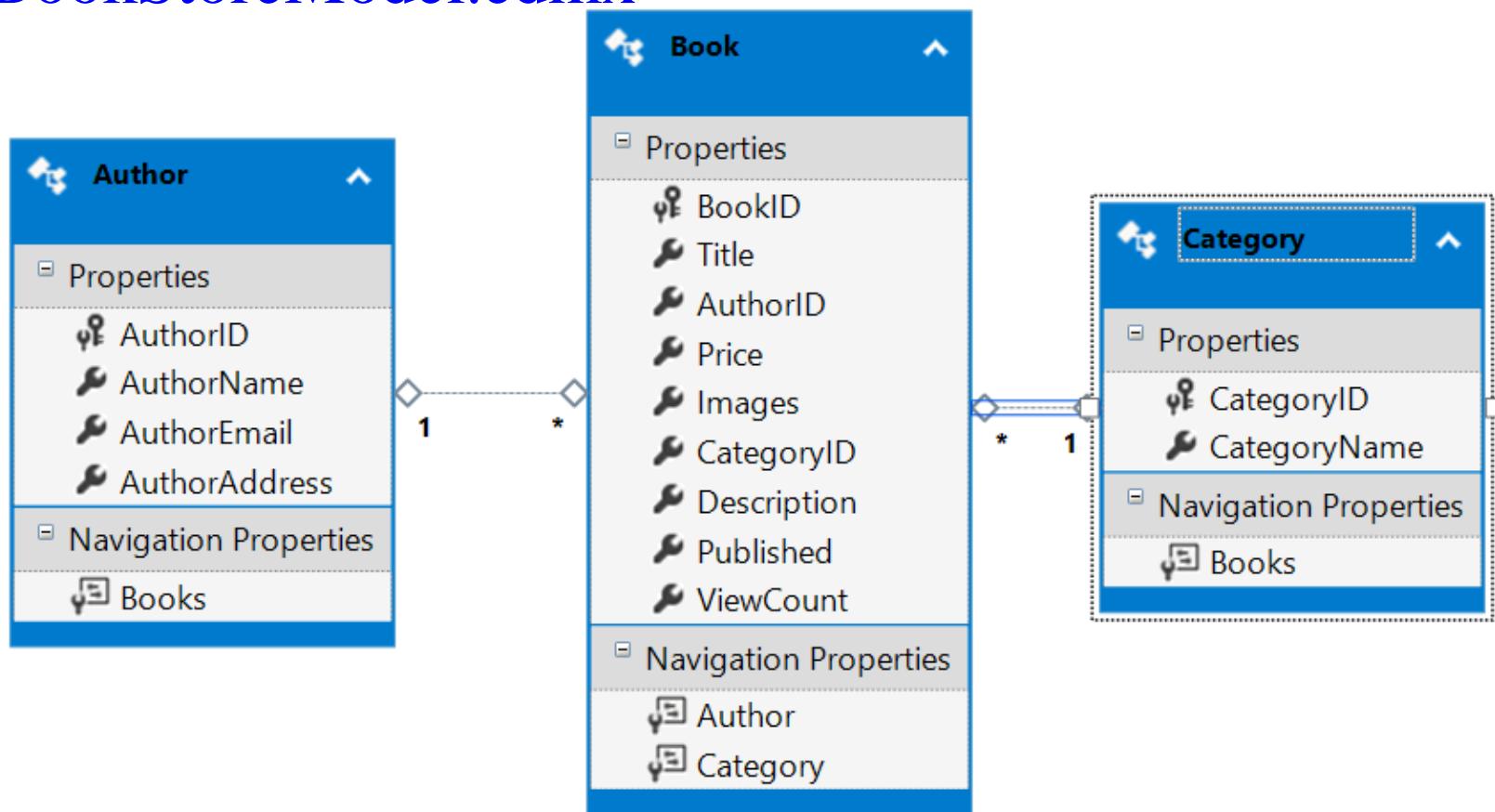
- ❖ Trong .NET Framework, **Data Annotation** dùng để thêm phần ý nghĩa mở rộng vào dữ liệu thông qua các thẻ thuộc tính.
- ❖ Tính năng **Data Annotation** được Microsoft giới thiệu lần đầu tiên ở phiên bản .NET 3.5 tại namespace **System.ComponentModel.DataAnnotations**. Namespace này chứa các lớp dùng để định nghĩa thuộc tính mở rộng cho dữ liệu.

3. CHÚ THÍCH DỮ LIỆU (Data Annotation)

- ❖ Các thuộc tính Data Annotation được phân chia thành 3 loại chính:
 - Thuộc tính xác nhận (Validation Attribute): dùng để thêm các tập luật xác nhận cho dữ liệu.
 - Thuộc tính hiển thị (Display Attribute): dùng để đặc tả cách dữ liệu từ một lớp được hiển thị ở giao diện.
 - Thuộc tính mô hình (Modelling Attribute): dùng để đặc tả mục đích sử dụng của lớp thành viên và mối quan hệ giữa các lớp.

3. CHÚ THÍCH DỮ LIỆU (Data Annotation)

- ❖ Sử dụng lại mô hình dữ liệu đã tạo trong phần trước BookStoreModel.edmx



3. CHÚ THÍCH DỮ LIỆU (Data Annotation)

❖ Tập tin Models/Book.cs (BookStoreModel.edmx/Book.cs):

```
namespace BookStoreManager.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Book
    {
        public int BookID { get; set; }
        public string Title { get; set; }
        public int AuthorID { get; set; }
        public Nullable<decimal> Price { get; set; }
        public string Images { get; set; }
        public int CategoryID { get; set; }
        public string Description { get; set; }
        public Nullable<System.DateTime> Published { get; set; }
        public Nullable<int> ViewCount { get; set; }

        public virtual Author Author { get; set; }
        public virtual Category Category { get; set; }
    }
}
```

3. CHÚ THÍCH DỮ LIỆU (Data Annotation)

- ❖ Chúng ta tiến hành thêm các chú thích dữ liệu cho các thuộc tính trong lớp Book.cs.
- ❖ Trước hết chúng ta cần phải thêm hai dòng sau để nhúng namespace DataAnnotations vào lớp Book.

```
using System.ComponentModel.DataAnnotations;  
using System.ComponentModel;
```

```

namespace BookStoreManager.Models
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel;
    using System.ComponentModel.DataAnnotations;

    public partial class Book
    {
        public int BookID { get; set; }

        [Required(ErrorMessage = "Xin vui lòng nhập sách"), MaxLength(100,
        ErrorMessage = "Tên sách không dài hơn 100 ký tự"), DisplayName("Tên sách")]

        public string Title { get; set; }

        [Required(ErrorMessage = "Xin vui lòng chọn tác giả"), DisplayName("Tác giả")]
        public int AuthorID { get; set; }

        [DisplayName("Đơn giá")]
        public Nullable<decimal> Price { get; set; }

        [DisplayName("Hình ảnh")]
        public string Images { get; set; }

        [Required(ErrorMessage = "Xin vui lòng chọn chủ đề sách"), DisplayName("Chủ đề sách")]
        public int CategoryID { get; set; }

        [DisplayName("Mô tả")]
        public string Description { get; set; }

        [DisplayName("Ngày xuất bản"), DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]
        public Nullable<System.DateTime> Published { get; set; }

        [DisplayName("Số lần xem")]
        public Nullable<int> ViewCount { get; set; }

        public virtual Author Author { get; set; }

        public virtual Category Category { get; set; }
    }
}

```

3. CHÚ THÍCH DỮ LIỆU (Data Annotation)

- ❖ Chúng ta có thể định nghĩa kiểu dữ liệu cho thuộc tính với các loại chú thích như sau:

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

3. CHÚ THÍCH DỮ LIỆU (Data Annotation)

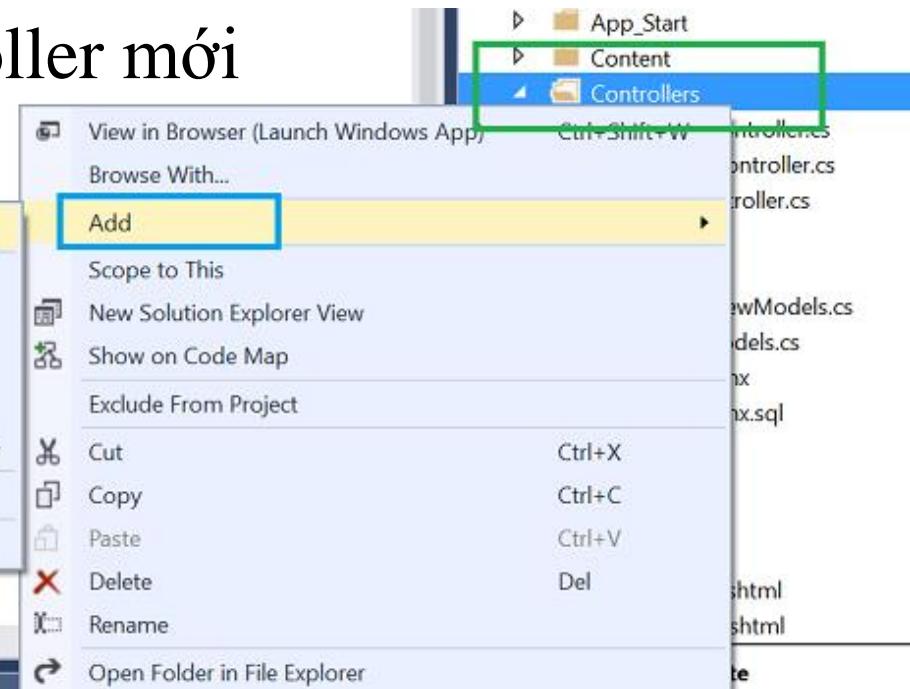
- ❖ Chúng ta có thể định nghĩa kiểu dữ liệu cho thuộc tính với các loại chú thích như sau:

Thuộc tính	Giải thích
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?
DisplayName	Thuộc tính hiển thị

4. Truy cập mô hình dữ liệu từ Controller

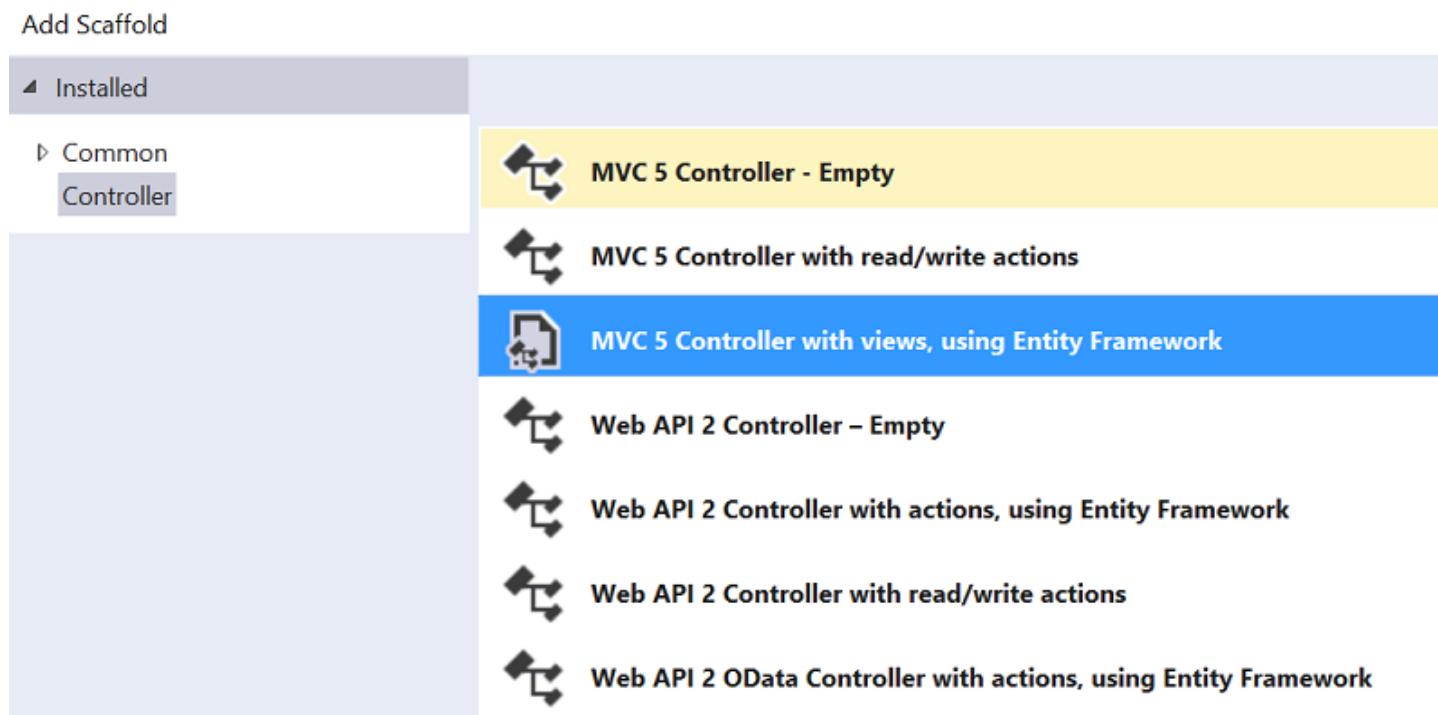
- ❖ Tạo lớp BookController mới và viết code để lấy dữ liệu từ bảng Book trong cơ sở dữ liệu và hiển thị ra trình duyệt sử dụng **view template**.
- ❖ **Bước 1:** Tạo 1 Controller mới

Controller



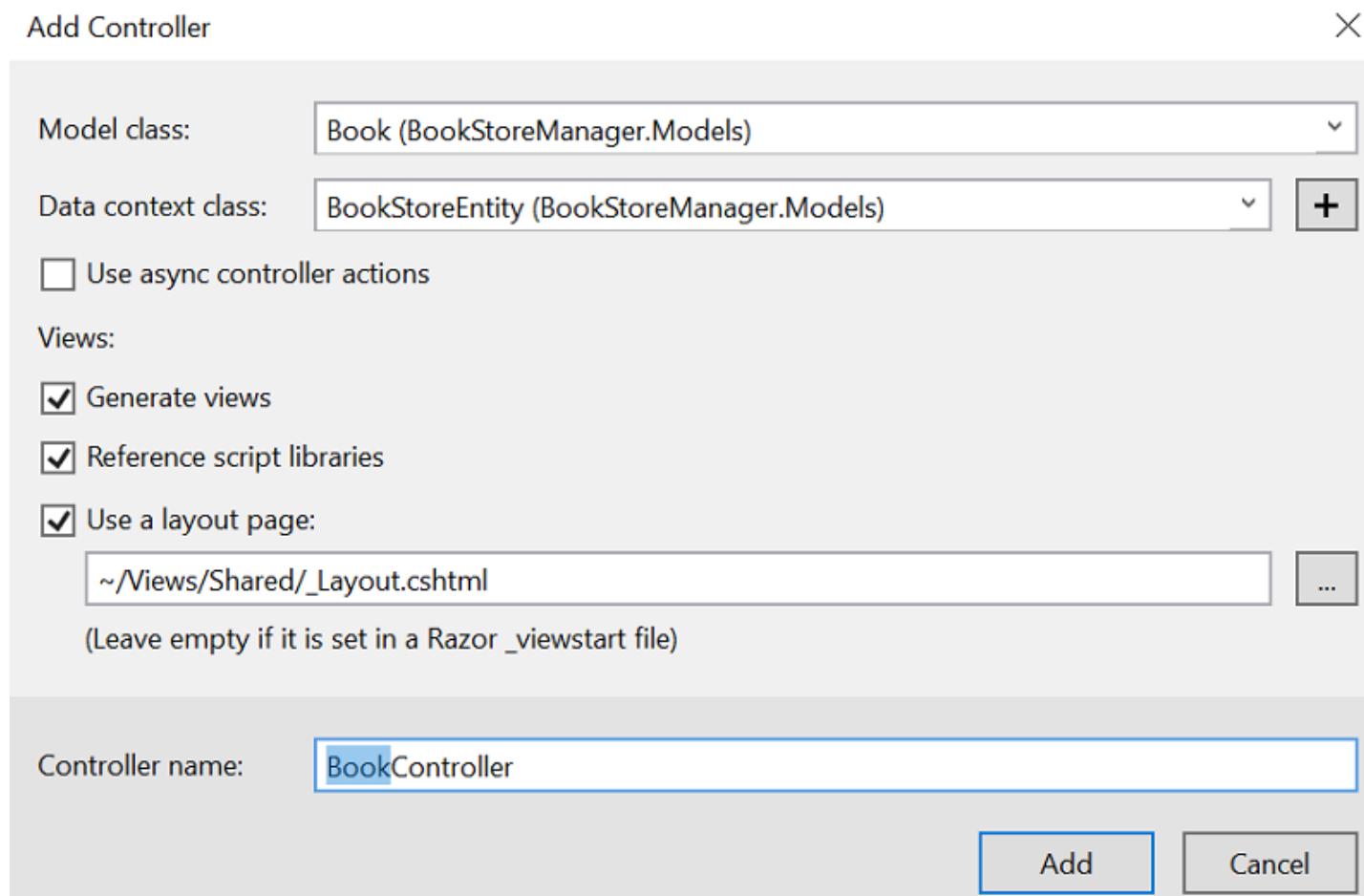
4. Truy cập mô hình dữ liệu từ Controller

❖ **Bước 2:** Ở hộp thoại Add Scaffold, chọn mục MVC 5 Controller with views (điều khiển MVC 5 với các view), sử dụng Entity Framework và nhấn Add.



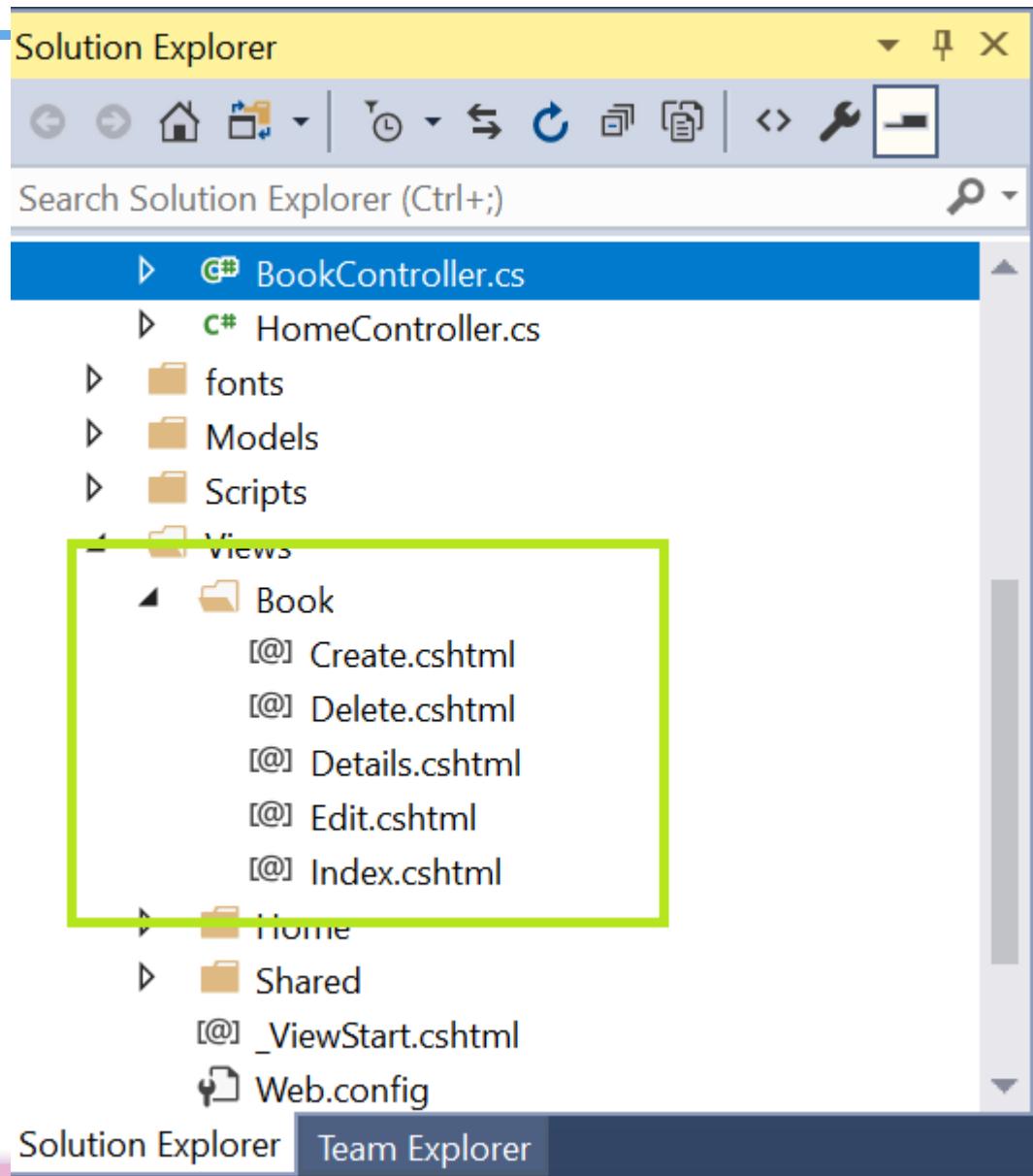
4. Truy cập mô hình dữ liệu từ Controller

❖ Bước 3:



4. Truy cập mô hình dữ liệu từ Controller

- ❖ Sau đó, Visual Studio sẽ tự động tạo mã nguồn các tập tin như hình kê bên:



4. Truy cập mô hình dữ liệu từ Controller

- ❖ Cấu hình lại tập tin **RouteConfig.cs** trong thư mục **App_Start** như sau:

```
1 reference
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Book", action = "Index", id = UrlParameter.Optional }
    );
}
```

4. Truy cập mô hình dữ liệu từ Controller

- Sau khi tạo xong BookController, chạy dự án web với đường dẫn <http://localhost:xxxx/Book/> để mở giao diện quản lý bảng Book mặc định.

Index							
Create New							
Tên sách	Đơn giá	Hình ảnh	Mô tả	Ngày xuất bán	Số lần xem	Tác giả	Chủ đề
Giáo trình Tin học cơ bản1	26000.00	THCB.jpg	Nội dung của cuốn: Tin Học Cơ Bản Windows XP gồm có 7 chương: Chương 1: Một số vấn đề cơ bản. Chương 2: Sử dụng nhanh công cụ và thanh thực đơn trong My Computer và Windows Explorer. Chương 3: Các thao tác trong windows XP. Chương 4: Các thiết lập trong Windows XP. Chương 5: Bảo trì máy tính. Chương 6: Các phím tắt Chương 7: Hỏi và đáp các thắc mắc. Xin trân trọng giới thiệu cuốn sách cùng bạn	17/07/2004 12:00:00 Sáng	0	Phạm Công Anh	Công nghệ thông tin Edit Details Delete
Giáo trình Tin học văn phòng	12000.00	TH004.jpg	Cuốn sách này gồm 3 phần sau: Phần 1: Xử lý văn bản trong Microsoft Office Word 2007. Phần 2: Xử lý bảng tính trong Microsoft Office Excel 2007. Phần 3: Quản lý cơ sở dữ liệu trong Microsoft Office Access 2007. Xin trân trọng giới thiệu cùng các bạn.	05/06/2004 12:00:00 Sáng	25	Phạm Công Anh	Văn học Edit Details Delete

5. Tùy chỉnh trang Index.cshtml

- ❖ Controllers\BookController.cs, phương thức Index được tạo tự động với đoạn mã như sau:

```
public class BookController : Controller
{
    private BookStoreEntity db = new BookStoreEntity();
    /
    public ActionResult Index()
    {
        var books = db.Books.Include(b => b.Author).Include(b => b.Category);
        return View(books.ToList());
    }
    ....
}
```

5. Tùy chỉnh trang Index.cshtml

Index.cshtml

```
Index.cshtml*  X
1  @model IEnumerable<Demo_Chuong06.Models.Book>
2  @{
3      ViewBag.Title = "Index";
4      Layout = "~/Views/Shared/_Layout.cshtml";
5  }
6  <h2>Index</h2>
7  <p>@Html.ActionLink("Create New", "Create")</p>
8  <table class="table">
9      <tr>
10         <th>
11             @Html.DisplayNameFor(model => model.Title)
12         </th>
13         <th>
14             @Html.DisplayNameFor(model => model.Price)
15         </th>
16         <th>
17             @Html.DisplayNameFor(model => model.Images)
18         </th>
19         <th>
20             @Html.DisplayNameFor(model => model.Description)
21         </th>
22         <th>
23             @Html.DisplayNameFor(model => model.Published)
24         </th>
25         <th>
26             @Html.DisplayNameFor(model => model.ViewCount)
27         </th>
28         <th>
29             @Html.DisplayNameFor(model => model.Author.AuthorName)
30         </th>
31         <th>
32             @Html.DisplayNameFor(model => model.Category.CategoryName)
33         </th>
34     <th></th>
```

5. Tùy chỉnh trang Index.cshtml

Index.cshtml

```
Index.cshtml*  X
35      </tr>
36      @foreach (var item in Model) {
37          <tr>
38              <td>
39                  @Html.DisplayFor(modelItem => item.Title)
40              </td>
41              <td>
42                  @Html.DisplayFor(modelItem => item.Price)
43              </td>
44              <td>
45                  @Html.DisplayFor(modelItem => item.Images)
46              </td>
47              <td>
48                  @Html.DisplayFor(modelItem => item.Description)
49              </td>
50              <td>
51                  @Html.DisplayFor(modelItem => item.Published)
52              </td>
53              <td>
54                  @Html.DisplayFor(modelItem => item.ViewCount)
55              </td>
56              <td>
57                  @Html.DisplayFor(modelItem => item.Author.AuthorName)
58              </td>
59              <td>
60                  @Html.DisplayFor(modelItem => item.Category.CategoryName)
61              </td>
62              <td>
63                  @Html.ActionLink("Edit", "Edit", new { id=item.BookID }) |
64                  @Html.ActionLink("Details", "Details", new { id=item.BookID }) |
65                  @Html.ActionLink("Delete", "Delete", new { id=item.BookID })
66              </td>
67          </tr>
68      }
```

5. Tùy chỉnh trang Index.cshtml

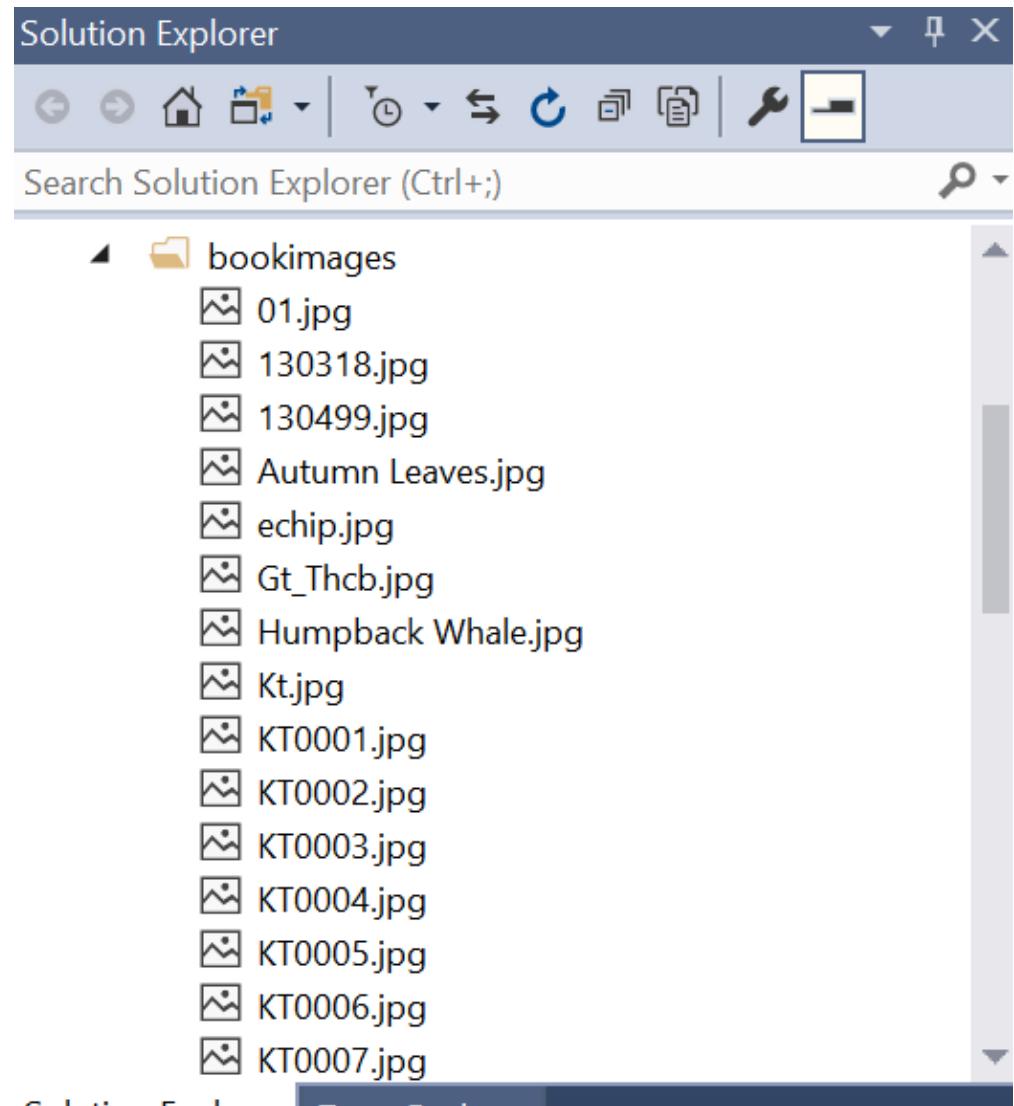
- ❖ Trong **Index.cshtml** có một số điều lưu ý sau:
 - Model của view này có kiểu là `@model IEnumerable<BookMan.Mvc.Models.Book>`. Chú ý rằng view model mà Index trả cho view có kiểu `Book[]` – mảng này thực thi `IEnumerable<Book>`.
 - Các lệnh có dạng `@Html.DisplayNameFor(model => model.Name)` được gọi là Html helper. Html helper là những phương thức C# hỗ trợ sinh HTML.

5. Tùy chỉnh trang Index.cshtml

- ❖ Tùy chỉnh một số chi tiết :
 1. Index -> Danh sách sách
 2. Create New -> Thêm sách mới
 3. Delete -> Xóa
 4. Details -> Xem
 5. Edit -> Sửa
 6. Xóa các cột: Mô tả, Số lần xem.
 7. Chính lại cột hình ảnh

5. Tùy chỉnh trang Index.cshtml

- ❖ Tùy chỉnh cột hình ảnh: Trước tiên tạo thư mục **bookimages** trong project, sau đó chuẩn bị các hình ảnh bên trong thư mục.



5. Tùy chỉnh trang Index.cshtml

- ❖ Tùy chỉnh thẻ sau:

```
    @Html.DisplayFor(modelItem => item.Images)
```

- ❖ Thành thẻ sau:

```
    <img src("~/bookimages/@item.Images" />
```

5. Tùy chỉnh trang Index.cshtml

- ❖ Thay đoạn code sau đây:

```
-<p>
| @Html.ActionLink("Create New", "Create")
|</p>
```

- ❖ Thành đoạn code sau:

```
-<p>
| @Html.ActionLink("Thêm sách mới",
|   "Create", null, new { @class = "btn btn-warning" })
|</p>
```

5. Tùy chỉnh trang Index.cshtml

Application name						
Thêm sách mới						
Tên sách	Đơn giá	Hình ảnh	Ngày xuất bản	Tác giả	Chủ đề	Sửa Xem Xóa
Giáo trình Tin học cơ bản 1	26000.00		17/07/2004	Phạm Công Anh	Công nghệ thông tin	Sửa Xem Xóa
Giáo trình Tin học văn phòng	12000.00		05/08/2004	Phạm Công Anh	Văn học	Sửa Xem Xóa
Lập Trình Cơ Sở Dữ Liệu Với Visual Basic 2005 Và ADO.NET 2.0	11500.00		12/07/2004	Nguyễn Thế Giang	Văn học	Sửa Xem Xóa
Visual Basic 2005 Tập 3, Quyển 2: Lập Trình Web	20000.00		18/07/2004	Lê Việt Nhân	Khoa học kỹ thuật	Sửa Xem Xóa
Giáo trình Tin học quản lý 1	21000.00		18/06/2004	Phạm Công Anh	Văn học	Sửa Xem Xóa

Tìm hiểu phương thức Create và View Create

The screenshot shows a web browser window with the title "Create - My ASP.NET Application". The URL in the address bar is "localhost:51922/Book/Create". The page content is a form for creating a book, with fields for Name, AuthorID, Unit Price, Image, CategoryID, Description, Publication Date, and Number of Views. The "CategoryID" field has "Ngoại ngữ" selected. At the bottom right of the form is a "Create" button.

← → C ⓘ localhost:51922/Book/Create

Anh Dung Diep Nhu Thè mới Download Ebook Bí... Bí Quyết Giàu Có V... Download and Inst... Download and Inst...

Application name Home About Contact

Tên sách

AuthorID Phạm Công Anh

Đơn giá

Hình ảnh

CategoryID Ngoại ngữ

Mô tả

Ngày xuất bản

Số lần xem

Create

Back to List

© 2023 - My ASP.NET Application

54

Tìm hiểu phương thức Create và View Create

❖ Tùy chỉnh View:

- Sửa giao diện Tạo Sách mới: Views\Book\Create.cshtml

```
@using (Html.BeginForm("Create", "Book", FormMethod.Post,  
new { enctype = "multipart/form-data" }))
```

```
{
```

```
...
```

```
}
```

- Tiếp theo thay đổi Image thành kiểu type = “file”

```
[@*  
@Html.EditorFor(model => model.Images, new { htmlAttributes = new { @class = "form-control" } })  
*@  
  
@Html.TextBoxFor(model => model.Images, null, new { type = "file", @class = "form-control" })
```

Tìm hiểu phương thức Create và View Create

❖ Tùy chỉnh View:

- Thay đổi "AuthorID" trong HTML thành "Tác giả"
- Thay đổi "CategoryID" trong HTML thành "Chủ đề"

```
@Html.LabelFor(model => model.AuthorID, "Tác giả",
    htmlAttributes: new { @class = "control-label col-md-2" })
```

```
@Html.LabelFor(model => model.CategoryID, "Chủ đề",
    htmlAttributes: new { @class = "control-label col-md-2" })
```

Tìm hiểu phương thức Create và View Create

❖ Tùy chỉnh View:

- Thay đổi @Html.EditorFor thành @Html.TextAreaFor chô mô tả sách như sau:

```
[@*
@Html.EditorFor(model => model.Description,
    new { htmlAttributes = new { @class = "form-control" } })
*@
@Html.TextAreaFor(model => model.Description,
    new { htmlAttributes = new { @class = "form-control" } })
```

Tìm hiểu phương thức Create và View Create

❖ Tùy chỉnh View:

- Chèn FCKEditor cho mô tả:

```
<div class="form-group">
    @Html.LabelFor(model => model.Description,
                  htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.TextAreaFor(model => model.Description,
                           new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Description, "",
                                   new { @class = "text-danger" })
        <script>
            CKEDITOR.replace("Description")
        </script>
    </div>
</div>
```

Tìm hiểu phương thức Create và View Create

❖ Tùy chỉnh View:

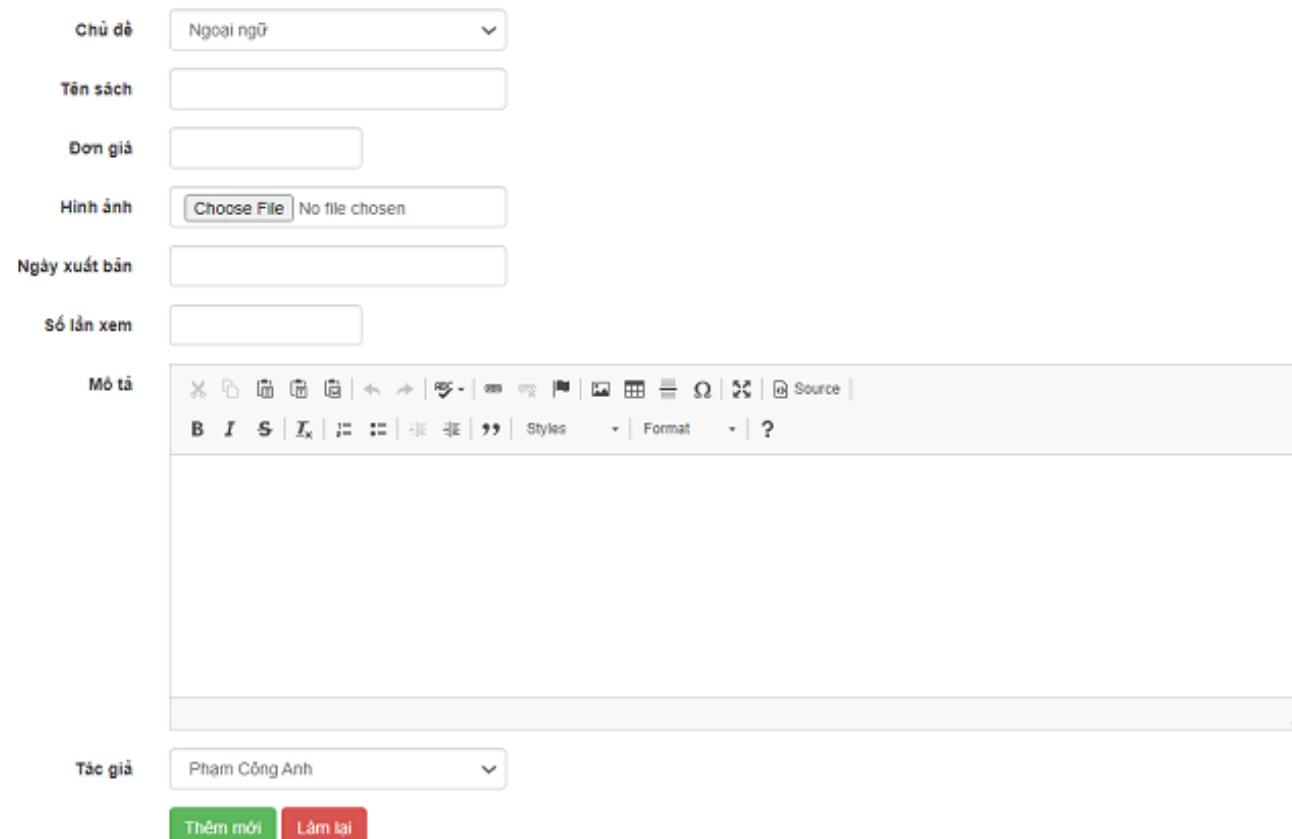
- Chèn thêm nút Reset vào kế bên nút submit:

```
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Thêm mới" class="btn btn-default" />
        <input type="reset" value="Làm lại" class="btn btn-danger" />
    </div>
</div>
```

Tìm hiểu phương thức Create và View Create

- ❖ Cuối cùng sắp xếp lại các dòng nhập cho hợp lý.

Thêm sách mới



Chủ đề: Ngoại ngữ

Tên sách:

Đơn giá:

Hình ảnh: Choose File No file chosen

Ngày xuất bản:

Số lần xem:

Mô tả:

Tác giả: Phạm Công Anh

Thêm mới | Làm lại

Tìm hiểu phương thức Edit và View Edit

❖ Tùy chỉnh View

- Các bước giống tùy chỉnh View Create.cshtml
- Trong View Edit ta thêm thẻ `` để hiển thị hình sách như sau:

```
<div class="form-group">
    @Html.LabelFor(model => model.Images, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Images, new { htmlAttributes = new { @class = "form-control" } })
        
        @Html.ValidationMessageFor(model => model.Images, "", new { @class = "text-danger" })
    </div>
</div>
```

Tìm hiểu phương thức Edit và View Edit

❖ Tùy chỉnh View

- Tiếp theo, trong View Edit ta thêm thẻ @Html.Hidden("oldimage", @Model.Images) để lưu ảnh:

```
<div class="form-group">
    @Html.LabelFor(model => model.Images, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Images,
            new { htmlAttributes = new { @class = "form-control" } })
        
        @Html.Hidden("oldimage", @Model.Images)
        @Html.ValidationMessageFor(model => model.Images, "", new { @class = "text-danger" })
    </div>
</div>
```

Chỉnh sửa sách

Chủ đề	Công nghệ thông tin
Tên sách	Giáo trình Tin học cơ bản1
Đơn giá	26000.00
Hình ảnh	<input type="button" value="Choose File"/> No file chosen 
Ngày xuất bản	17/07/2004
Số lần xem	1000
Mô tả	<p>Nội dung của cuốn: Tin Học Cơ Bản Windows XP gồm có 7 chương: Chương 1: Một số vấn đề cơ bản. Chương 2: Sử dụng nhanh thanh công cụ và thanh thực đơn trong My Computer và Windows Explorer. Chương 3: Các thao tác trong windows XP. Chương 4: Các thiết lập trong Windows XP. Chương 5: Bảo trì máy tính. Chương 6: Các phím tắt Chương 7: Hỏi và đáp các thắc mắc. Xin trân trọng giới thiệu cuốn sách cùng bạn</p>
Tác giả	Phạm Công Anh
	<input type="button" value="Thêm mới"/> <input type="button" value="Làm lại"/>

[Trở về danh sách](#)

Tìm hiểu phương thức Edit và View Edit

❖ Tùy chỉnh controller:

- Mở tập tin **BookController.cs**. Tiếp theo chúng ta sẽ xử lý trong Controller như sau, tìm đến hai phương thức **Edit()**:

```
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Book book = db.Books.Find(id);
    if (book == null)
    {
        return HttpNotFound();
    }
    ViewBag.AuthorID = new SelectList(db.Authors,
                                         "AuthorID", "AuthorName", book.AuthorID);
    ViewBag.CategoryID = new SelectList(db.Categories,
                                         "CategoryID", "CategoryName", book.CategoryID);
    return View(book);
}
```

Chỉnh sửa sách

Chủ đề	Công nghệ thông tin
Tên sách	Giáo trình Tin học cơ bản1
Đơn giá	26000.00
Hình ảnh	<input type="button" value="Choose File"/> No file chosen 
Ngày xuất bản	17/07/2004
Số lần xem	1000
Mô tả	<p>Nội dung của cuốn: Tin Học Cơ Bản Windows XP gồm có 7 chương: Chương 1: Một số vấn đề cơ bản. Chương 2: Sử dụng nhanh thanh công cụ và thanh thực đơn trong My Computer và Windows Explorer. Chương 3: Các thao tác trong windows XP. Chương 4: Các thiết lập trong Windows XP. Chương 5: Bảo trì máy tính. Chương 6: Các phím tắt Chương 7: Hỏi và đáp các thắc mắc. Xin trân trọng giới thiệu cuốn sách cùng bạn</p>
Tác giả	Phạm Công Anh

[Trở về danh sách](#)

Tìm hiểu phương thức Edit và View Edit

- ❖ Tùy chỉnh controller, sửa phương thức Edit post:

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "BookID,Title,AuthorID," +
    "Price,Images,CategoryID,Description,Published,ViewCount")] Book book)
{
    if (ModelState.IsValid)
    {
        db.Entry(book).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.AuthorID = new SelectList(db.Authors, "AuthorID",
        "AuthorName", book.AuthorID);
    ViewBag.CategoryID = new SelectList(db.Categories,
        "CategoryID", "CategoryName", book.CategoryID);
    return View(book);
}
```

```
[HttpPost]
[ValidateAntiForgeryToken]
[ValidateInput(false)]
public ActionResult Edit([Bind(Include = "BookID,Title,AuthorID,Price,Images," +
    "CategoryID,Description,Published,ViewCount")] Book book, HttpPostedFileBase Images, FormCollection form)
{
    if (ModelState.IsValid){
        try{
            if (Images != null){
                string _fileName = Path.GetFileName(Images.FileName);
                string _path = Path.Combine(Server.MapPath("~/bookimages"), _fileName);
                Images.SaveAs(_path);
                book.Images = _fileName;
                // get Path of old image for deleting it
                _path = Path.Combine(Server.MapPath("~/bookimages"), form["oldimage"]);
                if [System.IO.File.Exists(_path)]
                    System.IO.File.Delete(_path);

            }
            else
                book.Images = form["oldimage"];
            db.Entry(book).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        catch{
            ViewBag.Message = "không thành công!!";
        }
        return RedirectToAction("Index");
    }
    ViewBag.AuthorID = new SelectList(db.Authors, "AuthorID", "AuthorName", book.AuthorID);
    ViewBag.CategoryID = new SelectList(db.Categories, "CategoryID", "CategoryName", book.CategoryID);
    return View(book);
}
```

Tìm hiểu phương thức Edit và View Edit

- ❖ `ValidateAntiForgeryToken` kiểm chứng token XSRF (Cross-site request forgery) được tạo bởi `@Html.AntiForgeryToken()` ở view.
- ❖ Mô hình **ASP.NET MVC** lấy các giá trị cần post ở form, sau đó tạo một đối tượng **Book**, đẩy các giá trị này vào và kiểm chứng dữ liệu bằng phương thức `ModelState.IsValid`
- ❖ Đầu tiên là các tham số truyền vào trong **Action**, tham số book chính là các giá trị trong form được binding vào đối tượng, bao gồm BookID, Title, AuthorID,... còn tham số thứ 2 chính là phần ảnh mình upload lên cùng trong form, ASP.NET hỗ trợ kiểu `HttpPostedFileBase` để nhận dạng file upload. `FormCollection` để lấy hình lưu tạm tên `oldimage`.

Tìm hiểu phương thức Edit và View Edit

- ❖ Khi upload hình mới sẽ xóa hình cũ bằng lệnh sau `System.IO.File.Delete(_path)`. Trước khi xóa kiểm tra xem có hình trong thư mục? bằng lệnh `System.IO.File.Exists(_path)`
- ❖ Đồng thời mình lưu file vào trong thư mục /bookimages của project.
- ❖ Sau khi thực hiện xong, mình sẽ dùng `BookStoreEntity` để lưu dữ liệu vào trong CSDL.
- ❖ Dữ liệu `Book` được lưu vào database thông qua phương thức `SaveChanges()`.
- ❖ Trong trường hợp upload hình hoặc lưu dữ liệu không thành công thì chúng ta hiển thị thông báo lỗi thông qua `ViewBag.Message = "không thành công!!";`

Tìm hiểu phương thức Delete Và View Delete

❖ Tùy chỉnh lại View:

- Từ trang **Index.cshtml**. Sau đó, nhập liên kết **Xóa** trên giao diện Web để đi đến trang xóa với liên kết <http://localhost:xxxx/Book/Delete/13>:

Delete

Are you sure you want to delete this?

Book

Tên sách	Giáo trình Tin học cơ bản
Đơn giá	26000.00
Hình ảnh	TH004.jpg
Mô tả	<p>Nội dung của cuốn: Tin Học Cơ Bản Windows XP gồm có 7 chương: Chương 1: Một số vấn đề cơ bản. Chương 2: Sử dụng nhanh thanh công cụ và thanh thực đơn trong My Computer và Windows Explorer. Chương 3: Các thao t´c trong windows XP. Chương 4: Các thiết lập trong Windows XP. Chương 5: Bảo trì máy tín. Chương 6: Các phím tắt Chương 7: Hồi và d&aaacute;p các thắc mắc. Xin trân trong giới thiệu cuốn sách cùng ban</p>
Ngày xuất bản	02/12/2021
Số lần xem	100
Tác giả	Phạm Công Anh
Chủ đề	Công nghệ thông tin

[Delete](#)

| [Back to List](#)

Tìm hiểu phương thức Delete Và View Delete

❖ Tùy chỉnh lại View:

- Ngoài việc tùy chỉnh tiếng Anh thành tiếng Việt cho Form xóa ta còn tùy chỉnh một số thẻ sau:
- Để hiển thị hình chúng ta sẽ thay đổi
`@Html.DisplayFor(model => model.Images)` thành
``

```
<dd>
    
</dd>
```

Tìm hiểu phương thức Delete Và View Delete

❖ Tùy chỉnh lại View:

- Ở phần mô tả nội dung hiển thị luôn các thẻ html. Chúng ta sẽ đổi `@Html.DisplayFor(model => model.Description)` thành `@Html.Raw(Model.Description)`

```
<dd>
    @Html.Raw(Model.Description)
</dd>
```

Tìm hiểu phương thức Delete Và View Delete

❖ Tùy chỉnh lại View:

- Trong đoạn mã, bạn có thể xem giao diện trước khi xóa để xác nhận xóa. Bạn có thể thay đổi thông điệp “Bạn có muốn xóa quyền sách này?” nếu muốn. Tương tự các phương thức khác, ở View Delete cũng dùng `@Html.AntiForgeryToken()` để tạo 1 token bảo mật cho thao tác xóa.

Tìm hiểu phương thức Delete Và View Delete

❖ Kết quả sau khi tùy chỉnh Delete.cshtml:

Bạn có muốn xóa quyền sách này?

Tên sách	Giáo trình Tin học cơ bản
Đơn giá	26000.00
Hình ảnh	
Mô tả	Nội dung của cuốn: Tin Học Cơ Bản Windows XP gồm có 7 chương: Chương 1: Một số vấn đề cơ bản. Chương 2: Sử dụng nhanh thanh công cụ và thanh thực đơn trong My Computer và Windows Explorer. Chương 3: Các thao tác trong windows XP. Chương 4: Các thiết lập trong Windows XP. Chương 5: Bảo trì máy tính. Chương 6: Các phím tắt Chương 7: Hỏi và đáp các thắc mắc. Xin trân trọng giới thiệu cuốn sách cùng bạn
Ngày xuất bản	02/12/2021
Số lần xem	100
Tác giả	Phạm Công Anh
Chủ đề	Công nghệ thông tin

Xóa

| Quay về danh sách

Tìm hiểu phương thức Delete Và View Delete

❖ Phương thức Delete:

- Mở tập tin BookController.cs. Tiếp theo chúng ta sẽ xử lý trong Controller như sau, tìm đến hai phương thức Delete()
 - ✓ Phương thức Delete dạng GET
 - ✓ Phương thức Delete dạng POST

Tìm hiểu phương thức Delete Và View Delete

❖ Phương thức Delete dạng GET

```
// GET: Book/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Book book = db.Books.Find(id);
    if (book == null)
    {
        return HttpNotFound();
    }
    return View(book);
}
```

Tìm hiểu phương thức Delete Và View Delete

❖ Phương thức Delete dạng POST

```
// POST: Book/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Book book = db.Books.Find(id);
    db.Books.Remove(book);
    db.SaveChanges();
    return RedirectToAction("Index");
}
```

Tìm hiểu phương thức Details Và View Details

❖ Tùy chỉnh lại View:

- Từ trang **Index.cshtml**. Sau đó, nhập liên kết **Xem** trên giao diện Web để đi đến trang xem chi tiết với liên kết <http://localhost:xxxx/Book/Details/13>:

The screenshot shows a web application interface. At the top, there is a navigation bar with links: Application name, Home, About, and Contact. Below the navigation bar, the page title is "Details". Underneath the title, the word "Book" is displayed. A horizontal line separates this from the main content area. In the content area, there is a table-like structure with the following data:

Tên sách	Giáo trình Tin học cơ bản
Đơn giá	26000.00
Hình ảnh	TH004.jpg
Mô tả	<p>Nội dung của cuốn: Tin Học Cơ Bản Windows XP gồm có 7 chương: Chương 1: Một số vấn đề cơ bản. Chương 2: Sử dụng nhanh thanh công cụ và thanh thực đơn trong My Computer và Windows Explorer. Chương 3: Các thao tác trong windows XP. Chương 4: Các thiết lập trong Windows XP. Chương 5: Bảo trì mác y tính. Chương 6: Các phím tắt Chương 7: Hỏi và đác p các thắc mắc. Xin trân trọng giới thiệu cuốn sách cùng bạn</p>
Ngày xuất bản	02/12/2021
Số lần xem	100
Tác giả	Phạm Công Anh
Chú đề	Công nghệ thông tin

At the bottom left of the content area, there are two links: "Edit" and "Back to List".

Tìm hiểu phương thức Details Và View Details

❖ Tùy chỉnh lại View:

- Để hiển thị hình chúng ta sẽ thay đổi
`@Html.DisplayFor(model => model.Images)` thành
``

```
<dd>
    
</dd>
```

- Ở phần mô tả nội dung hiển thị luôn các thẻ html. Chúng ta sẽ đổi `@Html.DisplayFor(model => model.Description)` thành `@Html.Raw(Model.Description)`

```
<dd>
    @Html.Raw(Model.Description)
</dd>
```

Tìm hiểu phương thức Details Và View Details

❖ Tùy chỉnh lại View:

```
|<p>
|    @Html.ActionLink("Sửa", "Edit",| new { id = Model.BookID }) |
|    @Html.ActionLink("Quay trở lại danh sách", "Index")
</p>
```

Application name Home About Contact

Thông tin chi tiết sách

Tên sách	Giáo trình Tin học cơ bản
Đơn giá	26000.00
Hình ảnh	
Mô tả	Nội dung của cuốn: Tin Học Cơ Bản Windows XP gồm có 7 chương: Chương 1: Một số vấn đề cơ bản. Chương 2: Sử dụng nhanh thanh công cụ và thanh thực đơn trong My Computer và Windows Explorer. Chương 3: Các thao tác trong windows XP. Chương 4: Các thiết lập trong Windows XP. Chương 5: Bảo trì máy tính. Chương 6: Các phím tắt Chương 7: Hỏi và đáp các thắc mắc. Xin trân trọng giới thiệu cuốn sách cùng bạn!
Ngày xuất bản	02/12/2021
Số lần xem	100
Tác giả	Phạm Công Anh
Chủ đề	Công nghệ thông tin

Sửa | Quay trở lại danh sách

Tìm hiểu phương thức Details Và View Details

❖ Phương thức Details:

- Mở tập tin BookController.cs. Tiếp theo chúng ta sẽ xử lý trong Controller, tìm đến phương thức Details(int? id):

```
// GET: Book/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Book book = db.Books.Find(id);
    if (book == null)
    {
        return HttpNotFound();
    }
    return View(book);
}
```

Thêm phương thức Search và View tương ứng

- ❖ Controllers/BookController.cs, chỉnh sửa phương thức Index như sau:

```
// GET: Book
//public ActionResult Index()
//{
//    var books = db.Books.Include(b => b.Author).Include(b => b.Category);
//    return View(books.ToList());
//}
public ActionResult Index(string searchString)
{
    var books = db.Books.Include(b => b.Author).Include(b => b.Category);
    if (!String.IsNullOrEmpty(searchString))
    {
        searchString = searchString.ToLower();
        books = books.Where(b => b.Title.ToLower().Contains(searchString));
    }
    return View(books.ToList());
}
```

Thêm phương thức Search và View tương ứng

- ❖ Chạy chương trình với đường dẫn như sau:

<http://localhost:xxxx/Book?searchString=tin>

The screenshot shows a browser window with the URL localhost:60269/Book/Index?searchString=tin highlighted in red. The page title is "Danh sách sách". A button labeled "Thêm sách mới" is visible. The table displays four book entries, each with a "Sửa | Xem | Xóa" link.

Tên sách	Đơn giá	Hình ảnh	Ngày xuất bản	Tác giả	Chủ đề	Sửa Xem Xóa
Giáo trình <u>Tin</u> học cơ bản	26000.00		02/12/2021	Phạm Công Anh	Công nghệ thông tin	Sửa Xem Xóa
Giáo trình <u>Tin</u> học văn phòng	12000.00		05/06/2004	Nguyễn Thế Giang	Văn học	Sửa Xem Xóa
Giáo trình <u>Tin</u> học quản lý 1	21000.00		18/06/2004	Phạm Công Anh	Văn học	Sửa Xem Xóa
Giáo trình <u>Tin</u> học quản lý 2	21000.00		15/05/2004	Lê Việt Nhân	Công nghệ thông tin	Sửa Xem Xóa

Thêm phương thức Search và View tương ứng

❖ Tạo giao diện tìm kiếm ở View:

- Mở tập tin `Views\Book\Index.cshtml` và sau đoạn `@Html.ActionLink("Thêm sách mới", "Create", null, new { @class = "btn btn-warning" })`, thêm đoạn mã sau:

```
<!-- Đoạn cần thêm -->
@using (Html.BeginForm("Index", "Book", FormMethod.Get, new { @class = "form-inline" }))
{
    <div class="form-group">
        <label>Tên sách: </label>
        <input type="Text" class="form-control mx-sm-3" value="@ViewBag.Keyword"
               name="searchString" placeholder="Nhập tên sách cần tìm">
        <input type="button" class="btn btn-danger" value="Tìm sách">
    </div><hr />
}
<!-- Kết thúc -->
```

Thêm phương thức Search và View tương ứng

Application name Home About Contact

Danh sách sách

Thêm sách mới

Tên sách:

Tên sách	Đơn giá	Hình ảnh	Ngày xuất bản	Tác giả	Chủ đề	
Giáo trình Tin học cơ bản	26000.00		02/12/2021	Phạm Công Anh	Công nghệ thông tin	Sửa Xem Xóa
Giáo trình Tin học văn phòng	12000.00		05/06/2004	Nguyễn Thế Giang	Văn học	Sửa Xem Xóa
Giáo trình Tin học quản lý 1	21000.00		18/06/2004	Phạm Công Anh	Văn học	Sửa Xem Xóa
Giáo trình Tin học quản lý 2	21000.00		15/05/2004	Lê Việt Nhân	Công nghệ thông tin	Sửa Xem Xóa

Thêm phương thức Search và View tương ứng

❖ Tìm kiếm theo thể loại (Category)

➤ Controllers/BookController.cs, sửa phương thức Index:

```
public ActionResult Index(string searchString, int categoryID = 0)
{
    // 1. Lưu từ khóa tìm kiếm
    ViewBag.Keyword = searchString;
    //2.Tạo câu truy vấn kết 3 bảng Book, Author, Category
    var books = db.Books.Include(b => b.Author).Include(b => b.Category);
    //3. Tìm kiếm theo searchString
    if (!String.IsNullOrEmpty(searchString))
        books = books.Where(b => b.Title.Contains(searchString));
    //4. Tìm kiếm theo CategoryID
    if (categoryID != 0)
        books = books.Where(c => c.CategoryID == categoryID);
    //5. Tạo danh sách danh mục để hiển thị ở giao diện View thông qua DropDownList
    // danh sách Category
    ViewBag.CategoryID = new SelectList(db.Categories, "CategoryID", "CategoryName");
    return View(books.ToList());
}
```

Thêm phương thức Search và View tương ứng

❖ Tìm kiếm theo thể loại (Category)

- Ở tập tin **Views/Book/Index.cshtml**, bạn thay đổi nội dung mã nguồn như sau:

```
<!-- Đoạn cần thêm -->
@using (Html.BeginForm("Index", "Book", FormMethod.Get, new { @class = "form-inline" }))
{
    <div class="form-group">
        <label>Chủ đề:</label>@Html.DropDownList("CategoryID", null,
            htmlAttributes: new { @class = "form-control" }) <label>Tên sách: </label>
        <input type="Text" class="form-control mx-sm-3" value="@ViewBag.Keyword"
            name="searchString" placeholder="Nhập tên sách cần tìm">
        <input type="submit" class="btn btn-danger" value="Tìm sách">
    </div><hr />
}
<!-- Kết thúc -->
```

Thêm phương thức Search và View tương ứng

❖ Tìm kiếm theo thể loại (Category)

Application name Home About Contact

Danh sách sách

[Thêm sách mới](#)

Chủ đề: Công nghệ thông tin ▾ Tên sách: Tin học [Tim sách](#)

Tên sách	Đơn giá	Hình ảnh	Ngày xuất bản	Tác giả	Chủ đề	Sửa Xem Xóa
Giáo trình <u>Tin học cơ bản</u>	26000.00		02/12/2021	Phạm Công Anh	<u>Công nghệ thông tin</u>	Sửa Xem Xóa
Giáo trình <u>Tin học quản lý 2</u>	21000.00		15/05/2004	Lê Việt Nhân	<u>Công nghệ thông tin</u>	Sửa Xem Xóa
<u>Tin Học Ứng Dụng: Thành Thảo Oracle 9i - Quản Trị Cơ Sở Dữ Liệu (Tập 1)</u>	700000.00		16/09/2003	Phạm Công Anh	<u>Công nghệ thông tin</u>	Sửa Xem Xóa

© 2021 - My ASP.NET Application

6. PHÂN TRANG TRONG ASP.NET MVC

- ❖ Phân trang (pagination, paging) là một yêu cầu rất quan trọng khi hiển thị số lượng dữ liệu lớn.
- ❖ Để phân trang cần cài đặt gói **PagedList.Mvc**

Application name Home About Contact

Danh sách sách

[Thêm sách mới](#)

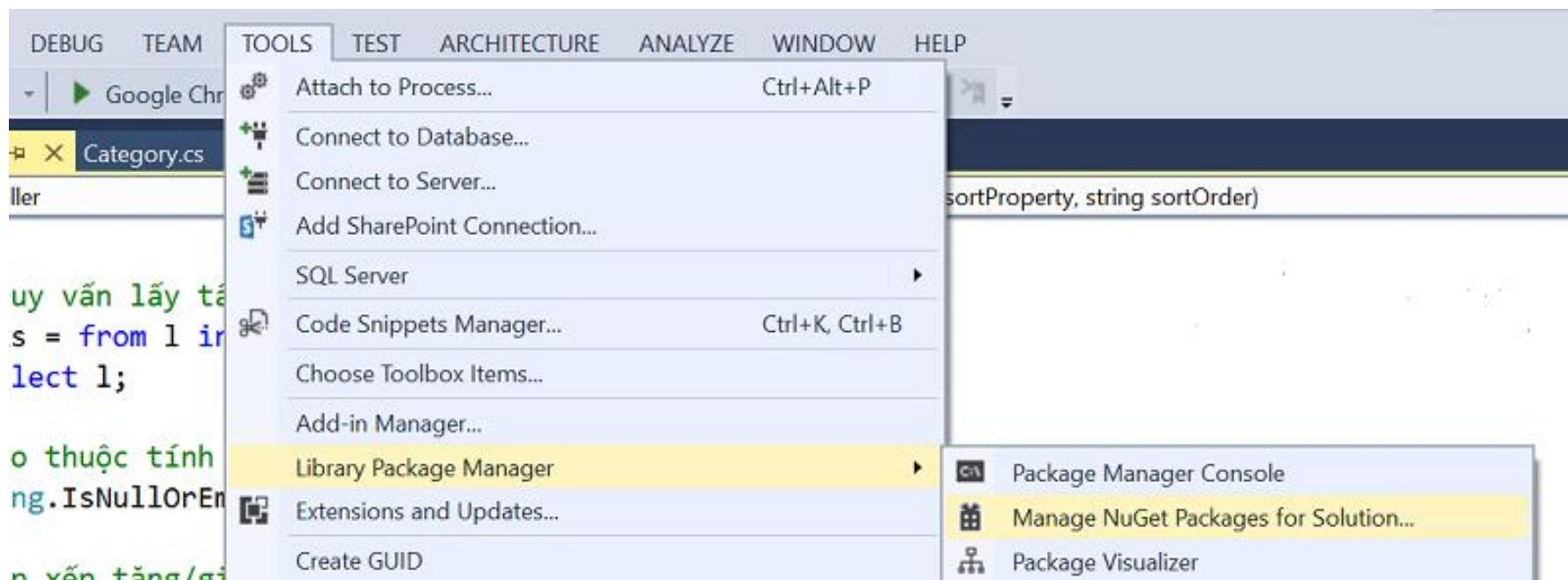
Tên sách	Chủ đề	Giá	Hình ảnh	Ngày xuất bản	Tác giả	Sửa Xem Xóa
Visual Basic 2005 Tập 3, Quyển 2: Lập Trình Web	Khoa học kỹ thuật	20000.00		18/07/2004	Lê Việt Nhân	Sửa Xem Xóa
Giáo trình Tin học quản lý 1	Văn học	21000.00		18/06/2004	Phạm Công Anh	Sửa Xem Xóa
Giáo trình Tin học quản lý 2	Công nghệ thông tin	21000.00		15/05/2004	Lê Việt Nhân	Sửa Xem Xóa

Trang 2 / 5

[«](#) [1](#) [2](#) [3](#) [4](#) [5](#) [»](#)

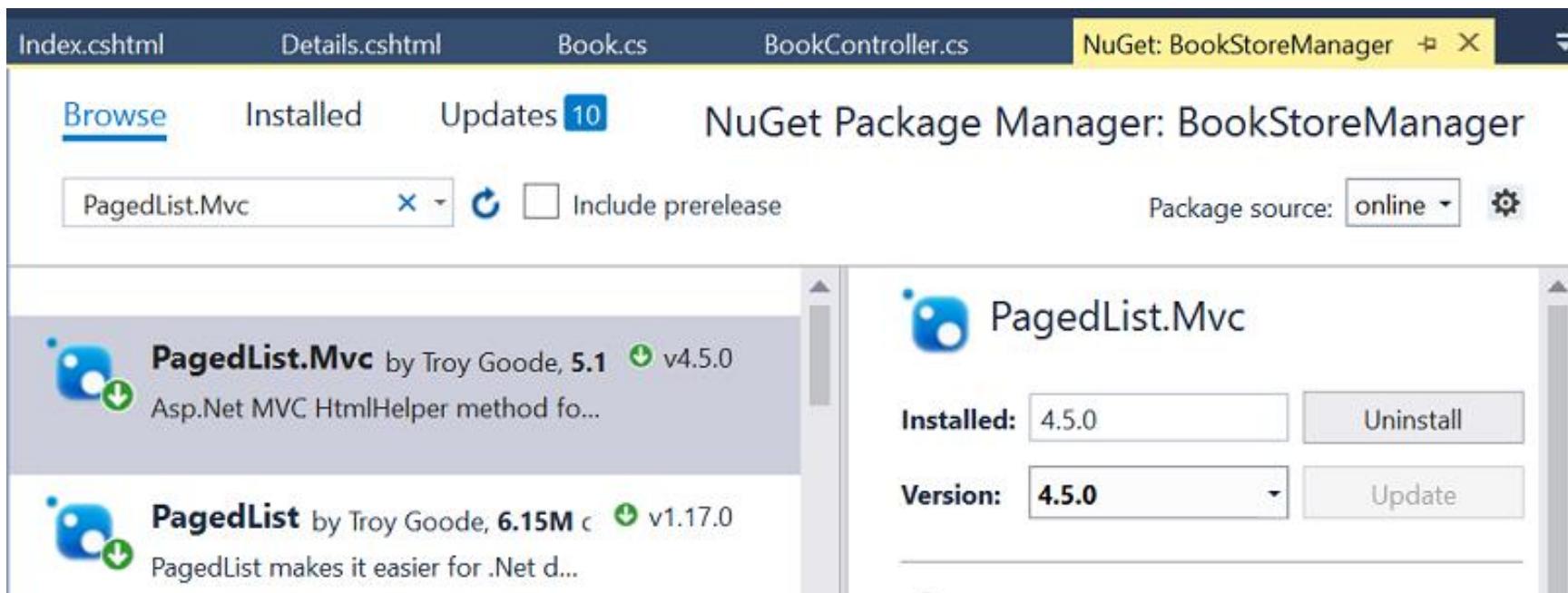
6. PHÂN TRANG TRONG ASP.NET MVC

- ❖ Ở giao diện Visual Studio bạn chọn mục Tools -> Library Package Manager -> Manage NuGet Packages... để thực hiện cài đặt.



6. PHÂN TRANG TRONG ASP.NET MVC

- ❖ Sau đó gõ **PagedList.Mvc** bên ô **Search** góc trái trên cùng, chọn gói **PagedList.Mvc** để cài đặt, nhấn nút **Install**.



6. PHÂN TRANG TRONG ASP.NET MVC

❖ Tiến hành phân trang:

```
using PagedList;
```

```
public ActionResult Index(int? page)
{
    // 1. Tham số int? dùng để thể hiện null và kiểu int
    // page có thể có giá trị là null và kiểu int.
    // 2. Nếu page = null thì đặt lại là 1.
    if (page == null) page = 1;
    // 3. Tạo truy vấn, lưu ý phải sắp xếp theo trường nào đó, ví dụ OrderBy
    // theo BookID mới có thể phân trang.
    var books = db.Books.Include(b=>b.Author).Include(b=>b.Category).OrderBy(b=>b.BookID);
    // 4. Tạo kích thước trang (pageSize) hay là số Link hiển thị trên 1 trang
    int pageSize = 3;
    // 4.1 Toán tử ?? trong C# mô tả nếu page khác null thì lấy giá trị page, còn
    // nếu page = null thì lấy giá trị 1 cho biến pageNumber.
    int pageNumber = (page ?? 1);
    // 5. Trả về các Link được phân trang theo kích thước và số trang.
    return View(books.ToPagedList(pageNumber, pageSize));
}
```

6. PHÂN TRANG TRONG ASP.NET MVC

❖ Toán tử ??

```
int pageNumber = (page ?? 1);
```

➤ Ví dụ:

```
int? x = null;  
int y = x ?? 1; // Kết quả y là 1, do x = null.
```

```
int? a = 5;  
int b = a ?? 1; // Kết quả b là 5 do a không phải là null.
```

6. PHÂN TRANG TRONG ASP.NET MVC

❖ Tiến hành phân trang:

➤ Sửa tập tin Views/Book/Index.cshtml

```
<!-- Thêm thư viện -->
@model PagedList.IagedList<Demo_Chuong06.Models.Book>
@using PagedList.Mvc;
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Danh sách sách</h2>
<p>
    @Html.ActionLink("Thêm sách mới", "Create", null, new { @class = "btn btn-warning" })
</p>

<table class="table">...</table>
<br />
<br />

Trang @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber)/@Model.PageCount
@Html.PagedListPager(Model, page => Url.Action("Index", new { page }))
<!-- Kết thúc --&gt;</pre>
```

6. PHÂN TRANG TRONG ASP.NET MVC

```
12  <!-- Sửa lại các cột tiêu đề --&gt;
13  &lt;table class="table"&gt;
14      &lt;tr&gt;
15          &lt;th style="width:200px"&gt;Tên sách&lt;/th&gt;
16          &lt;th&gt;Chủ đề&lt;/th&gt;
17          &lt;th&gt;Giá&lt;/th&gt;
18          &lt;th&gt;Hình ảnh&lt;/th&gt;
19          &lt;th&gt;Ngày xuất bản&lt;/th&gt;
20          &lt;th&gt;Tên tác giả&lt;/th&gt;
21          &lt;th&gt;Sửa | Xem | Xóa&lt;/th&gt;
22      &lt;/tr&gt;
23      @foreach (var item in Model)
24      {
25          &lt;tr&gt;
26              &lt;td&gt;
27                  @Html.DisplayFor(modelItem =&gt; item.Title)
28              &lt;/td&gt;
29              &lt;td&gt;
30                  @Html.DisplayFor(modelItem =&gt; item.Category.CategoryName)
31              &lt;/td&gt;</pre>
```

6. PHÂN TRANG TRONG ASP.NET MVC

```
32 <td>
33     @Html.DisplayFor(modelItem => item.Price)
34 </td>
35 <td>
36     
37 </td>
38 <td>
39     @Html.DisplayFor(modelItem => item.Published)
40 </td>
41 <td>
42     @Html.DisplayFor(modelItem => item.Author.AuthorName)
43 </td>
44 <td>
45     @Html.ActionLink("Sửa", "Edit", new { id = item.BookID }) |
46     @Html.ActionLink("Xem", "Details", new { id = item.BookID }) |
47     @Html.ActionLink("Xóa", "Delete", new { id = item.BookID })
48 </td>
49 </tr>
50 }
51 </table>
52 <br />
```

6. PHÂN TRANG TRONG ASP.NET MVC

- ❖ Chạy ứng dụng ta có kết quả:

Application name Home About Contact

Danh sách sách

[Thêm sách mới](#)

Tên sách	Chủ đề	Giá	Hình ảnh	Ngày xuất bản	Tên tác giả	Sửa Xem Xóa
Visual Basic 2005 Tập 3, Quyển 2: Lập Trình Web	Khoa học kỹ thuật	20000.00		18/07/2004	Lê Việt Nhân	Sửa Xem Xóa
Giáo trình Tin học quản lý 1	Văn học	21000.00		18/06/2004	Phạm Công Anh	Sửa Xem Xóa
Giáo trình Tin học quản lý 2	Công nghệ thông tin	21000.00		15/05/2004	Lê Việt Nhân	Sửa Xem Xóa

Trang 2 / 5

6. PHÂN TRANG TRONG ASP.NET MVC

❖ Phân trang nâng cao trong ASP.NET MVC:

Application name: Home About Contact

Danh sách sách

[Thêm sách mới](#)

Tên sách	Chủ đề	Giá	Hình ảnh	Ngày xuất bản	Tác giả	Sửa Xem Xóa
Giáo trình Tin học cơ bản	Công nghệ thông tin	26000.00		02/12/2021	Phạm Công Anh	Sửa Xem Xóa
Giáo trình Tin học văn phòng	Văn học	12000.00		05/06/2004	Nguyễn Thế Giang	Sửa Xem Xóa
Lập Trình Cơ Sở Dữ Liệu Với Visual Basic 2005 Và ADO.NET 2.0	Văn học	11500.00		12/07/2004	Nguyễn Thế Giang	Sửa Xem Xóa
Visual Basic 2005 Tập 3, Quyển 2: Lập Trình Web	Khoa học kỹ thuật	20000.00		18/07/2004	Lê Việt Nhân	Sửa Xem Xóa
Giáo trình Tin học quản lý 1	Văn học	21000.00		18/06/2004	Phạm Công Anh	Sửa Xem Xóa

Kích thước trang: 5

Trang: 1 / 3

1 2 3 4

6. PHÂN TRANG TRONG ASP.NET MVC

- ❖ Phân trang nâng cao trong ASP.NET MVC:
 - Mở tập tin **BookController.cs** và chỉnh sửa phương thức Index.
 - Trước hết, thêm 2 tham số **int? size, int? page** vào phương thức và thuộc tính **[HttpGet]** được thêm ở đầu phương thức Index dùng để định nghĩa việc submit form ở View là kiểu **HttpGet** nhằm lấy biến dữ liệu trên URL (ví dụ: <http://localhost:xxxx/Book?page=2&size=10>).
 - Chúng ta sử dụng một List với mỗi item kiểu **SelectListItem** để thêm giá trị kích thước trang.

6. PHÂN TRANG TRONG ASP.NET MVC

```
[HttpGet]
public ActionResult Index(int? size, int? page)
{
    // Bạn có thể thêm bớt tùy ý
    List<SelectListItem> items = new List<SelectListItem>();
    items.Add(new SelectListItem { Text = "3", Value = "3" });
    items.Add(new SelectListItem { Text = "5", Value = "5" });
    items.Add(new SelectListItem { Text = "10", Value = "10" });
    items.Add(new SelectListItem { Text = "25", Value = "25" });
    items.Add(new SelectListItem { Text = "50", Value = "50" });
    items.Add(new SelectListItem { Text = "100", Value = "100" });
    items.Add(new SelectListItem { Text = "200", Value = "200" });

    //1.1. Giữ trạng thái kích thước trang được chọn trên DropDownList
    foreach (var item in items)
        if (item.Value == size.ToString()) item.Selected = true;
    //1.2. Tạo các biến ViewBag
    ViewBag.size = items; // ViewBag DropDownList
    ViewBag.currentSize = size; // tạo biến kích thước trang hiện tại
```

6. PHÂN TRANG TRONG ASP.NET MVC

```
//2. Nếu page = null thì đặt lại là 1.  
page = page ?? 1; //if (page == null) page = 1;  
//3. Tạo truy vấn, lưu ý phải sắp xếp theo trường nào đó, ví dụ OrderBy  
//theo LinkID mới có thể phân trang.  
var books=db.Books.Include(b=>b.Author).Include(b=>b.Category).OrderBy(b=>b.BookID);  
  
//4. Tạo kích thước trang (pageSize) hay là số Link hiển thị trên 1 trang  
//Tạo kích thước trang (pageSize), mặc định là 5.  
int pageSize = (size ?? 3);  
// 4.1 Toán tử ?? trong C# mô tả nếu page khác null thì lấy giá trị page, còn  
// nếu page = null thì lấy giá trị 1 cho biến pageNumber.  
int pageNumber = (page ?? 1);  
// 5. Trả về các Link được phân trang theo kích thước và số trang.  
return View(books.ToPagedList(pageNumber, pageSize));  
}
```

6. PHÂN TRANG TRONG ASP.NET MVC

❖ Phân trang nâng cao trong ASP.NET MVC:

➤ Sửa giao diện Views/Book/Index.cshtml

```
<br />
<!-- Thêm mã phân trang -->
<br />
<div class="row">
    <div class="col-md-6 col-lg-6 col-xs-6 col-sm-6">
        @using (Html.BeginForm("Index", "Book", FormMethod.Get))
        {
            <p>
                Kích thước trang: @Html.DropDownList("size",
                    (List<SelectListItem>)ViewBag.size, new { @onchange = "this.form.submit();" })
            </p>
        }
    </div>
    <div class="col-md-6 col-lg-6 col-xs-6 col-sm-6 text-right">
        Trang: @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) / @Model.PageCount
    </div>
</div>

@Html.PagedListPager(Model, page => Url.Action("Index", new { page, size = ViewBag.currentSize }))
```

6. PHÂN TRANG TRONG ASP.NET MVC

❖ Phân trang nâng cao trong ASP.NET MVC:

Application name: Home About Contact

Danh sách sách

[Thêm sách mới](#)

Tên sách	Chủ đề	Giá	Hình ảnh	Ngày xuất bản	Tác giả	Sửa Xem Xóa
Giáo trình Tin học cơ bản	Công nghệ thông tin	26000.00		02/12/2021	Phạm Công Anh	Sửa Xem Xóa
Giáo trình Tin học văn phòng	Văn học	12000.00		05/06/2004	Nguyễn Thế Giang	Sửa Xem Xóa
Lập Trình Cơ Sở Dữ Liệu Với Visual Basic 2005 Và ADO.NET 2.0	Văn học	11500.00		12/07/2004	Nguyễn Thế Giang	Sửa Xem Xóa
Visual Basic 2005 Tập 3, Quyển 2: Lập Trình Web	Khoa học kỹ thuật	20000.00		18/07/2004	Lê Việt Nhẫn	Sửa Xem Xóa
Giáo trình Tin học quản lý 1	Văn học	21000.00		18/06/2004	Phạm Công Anh	Sửa Xem Xóa

Kích thước trang:

Trang: 1 / 3

1 2 3 4

