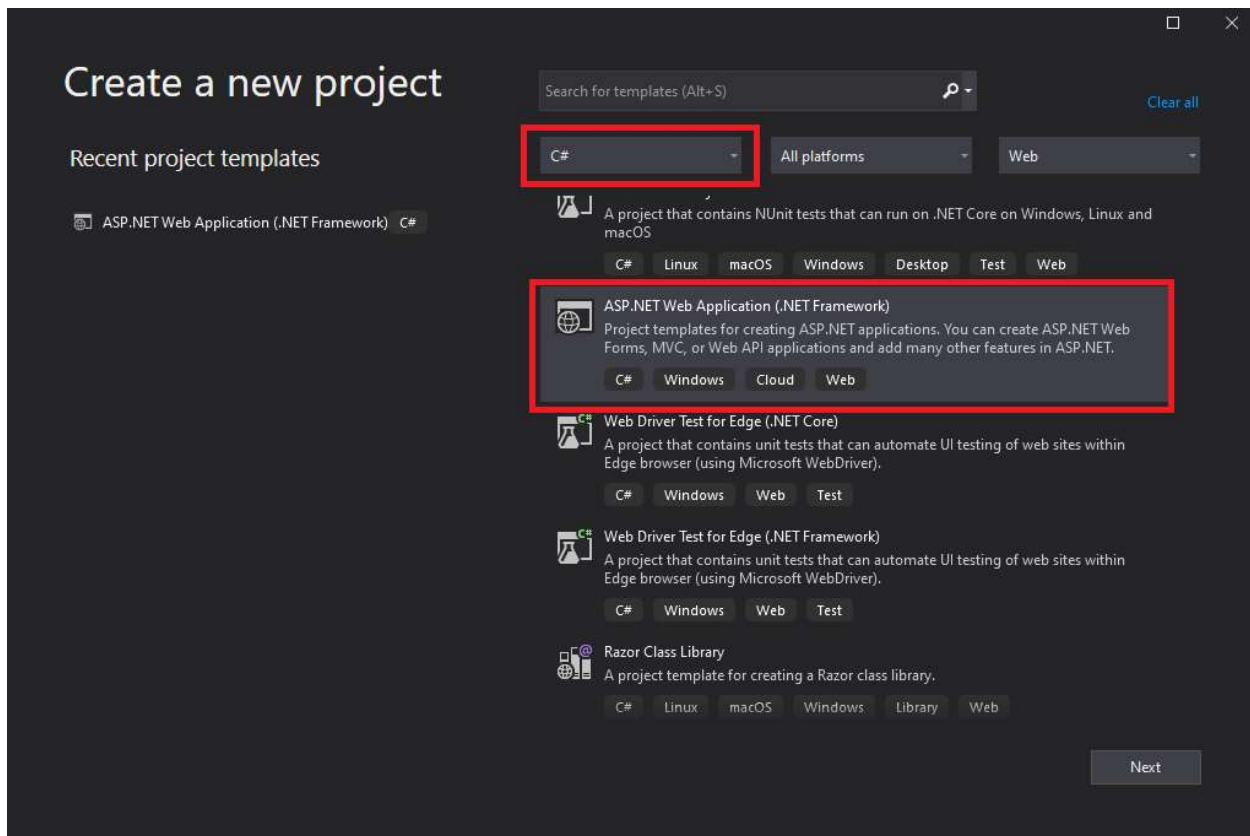


TẠO PROJECT QUẢN LÝ SÁCH (CÓ KẾT NỐI DATABASE)

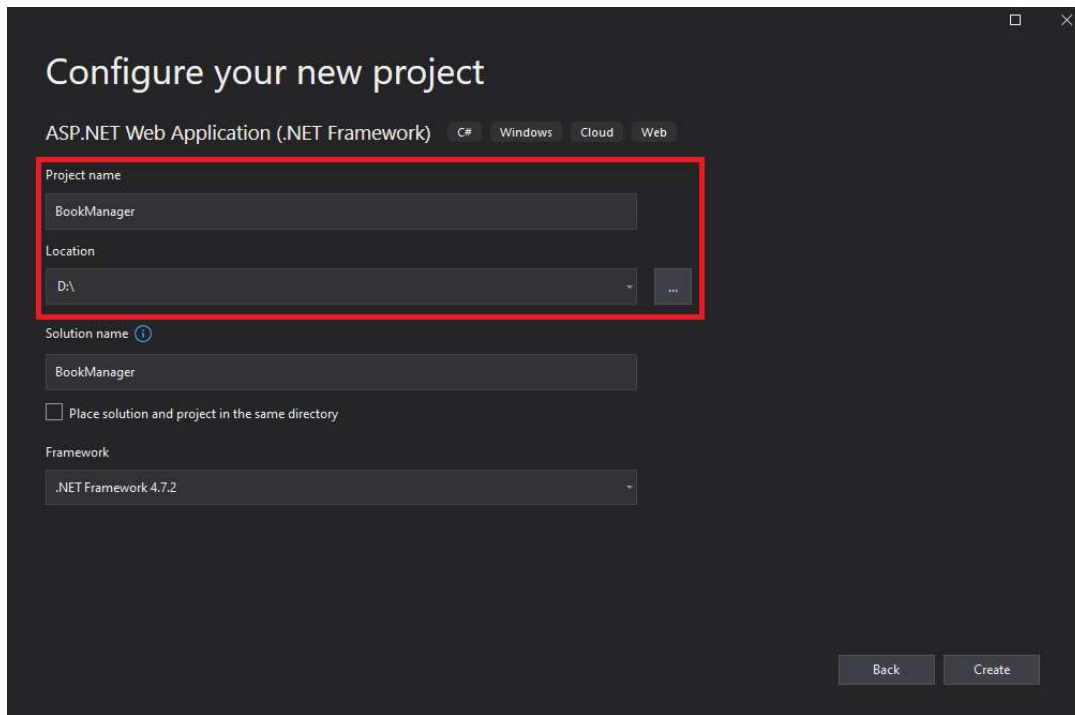
Trong bài lab này sử dụng IDE Visual Studio 2019 và Microsoft SQL Server 2019. Các phiên bản còn lại cũng có thao tác tương tự.

Bước 1: Tạo project ASP .NET MVC

Chọn **Create a new project** → chọn ngôn ngữ là **C#** → chọn **ASP .NET Web Application (.NET Framework)**



Nhấn **Next** → Tiến hành đặt tên và chọn đường dẫn lưu trữ project → Nhấn **Create**



Configure your new project

ASP.NET Web Application (.NET Framework) C# Windows Cloud Web

Project name
BookManager

Location
D:\

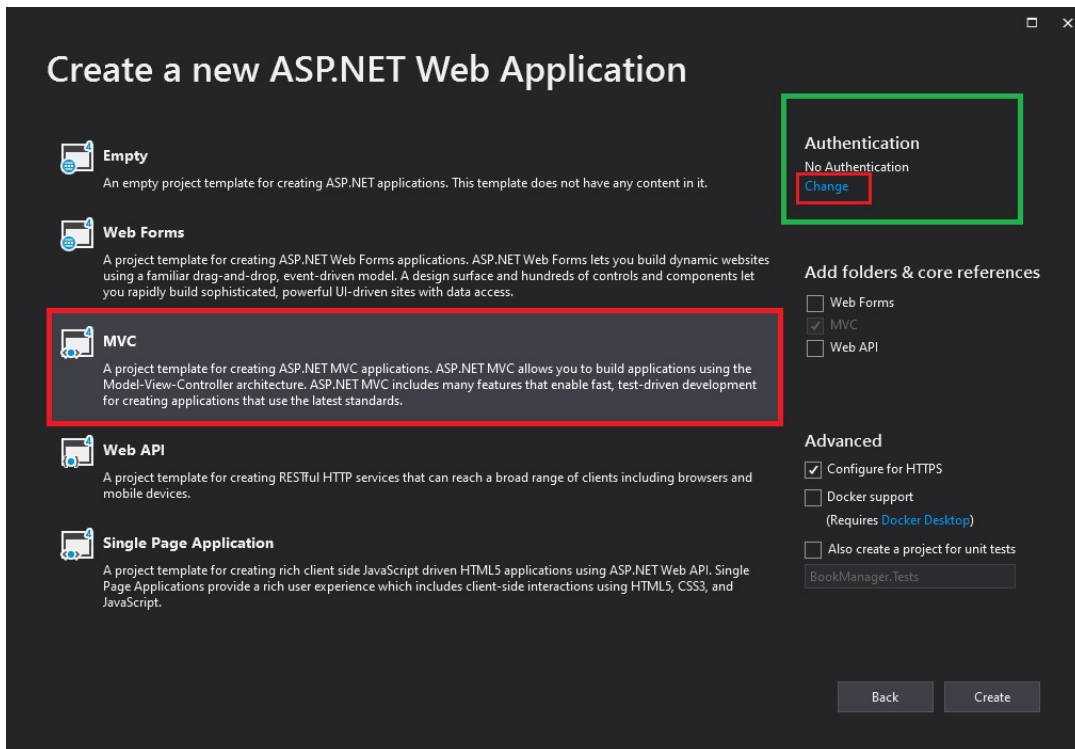
Solution name
BookManager

☐ Place solution and project in the same directory

Framework
.NET Framework 4.7.2

Back Create

Chọn template là **MVC**. Trong phần **Authentication** → Nhấn **Change**



Create a new ASP.NET Web Application

Empty
An empty project template for creating ASP.NET applications. This template does not have any content in it.

Web Forms
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

MVC
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Web API
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

Single Page Application
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication
No Authentication
[Change](#)

Add folders & core references

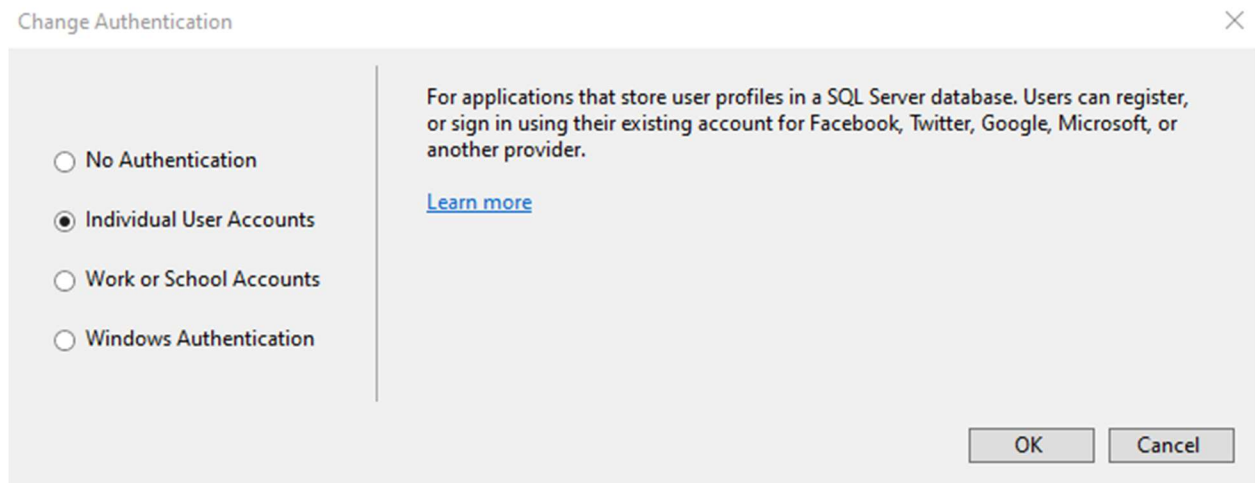
☐ Web Forms
☒ MVC
☐ Web API

Advanced

☒ Configure for HTTPS
☐ Docker support
(Requires [Docker Desktop](#))
☐ Also create a project for unit tests
BookManager.Tests

Back Create

Trong **Change Authentication**, chọn kiểu Authentication là **Individual User Accounts** và click OK để trở về cửa sổ trước đó và tiến hành **Create** project



Khi project được tạo xong các bạn sẽ thấy 1 số files được tạo sẵn trong project như sau:

- **App_Start/IdentityConfig.cs**: chứa các lệnh để cấu hình ASP.NET Identity
- **Controller/AccountController**: controller chứa các action method có tác dụng xác thực người dùng như Login, Register, ForgotPassword, ...
- **Controller/ManageController**: controller chứa các action method có tác dụng quản lý user (khi user đã login vào web) như ChangePassword, SetPassword, ...
- **Model/AccountViewModels**: chứa các View Model hiển thị trong các view của AccountController
- **Model/ManageViewModels**: chứa các View Model hiển thị trong các view của ManageController
- **Model/IdentityModels**: chứa class **ApplicationUser** để quản lý thông tin user và class **ApplicationDbContext** để quản lý kết nối với database ở dạng **Entity Framework Code First**

Bước 2: Sử dụng các chức năng đăng ký, đăng nhập,... được xây dựng sẵn khi sử dụng Authentication **Individual User Accounts**

Trước khi sử dụng các chức năng được xây dựng sẵn ở trên, chúng ta cần phải cấu hình chuỗi kết nối (**connectionString**) trong file **Web.config**. Bởi vì khi sử dụng các chức năng đó, Visual Studio sẽ tự tạo mới một database (nếu database chưa tồn tại) để chứa các table của ASP.NET Identity hoặc add thêm các table của ASP.NET Identity vào database (nếu database đó đã tồn tại) trong SQL Server. Tùy theo server SQL Server và database của mình là gì thì chúng ta cấu hình trong file **Web.config** cho phù hợp. Giả sử, ở đây chúng ta muốn kết nối đến server **DESKTOP-19216HB\SQLEXPRESS** với database tên là **BookManager** (database này chúng ta sẽ không tạo trước vì Visual Studio sẽ tự tạo khi ta sử dụng bất kỳ chức năng nào được xây dựng sẵn như đăng ký, đăng nhập,...).

Thay đổi đoạn **connectString** như sau:

Data Source: là tên Server name của MS SQL

Bỏ đi dòng **AttachDbFilename**

Initial Catalog: là tên database muốn tạo

Các dòng còn lại giữ nguyên

```
<configSections>
  <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
  <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
</configSections>
<connectionStrings>
  <add name="DefaultConnection" connectionString="Data Source=DESKTOP-19216HB\SQLEXPRESS;Initial Catalog=BookManager;Integrated Security=True" providerName="System.Data.SqlClient" />
</connectionStrings>
<appSettings>
  <add key="webpages:Version" value="3.0.0.0" />
  <add key="webpages:Enabled" value="false" />
</appSettings>
```

```
<connectionStrings>
  <add name="DefaultConnection" connectionString="Data Source=DESKTOP-19216HB\SQLEXPRESS;Initial Catalog=BookManager;Integrated Security=True" providerName="System.Data.SqlClient" />
</connectionStrings>
```

Sau khi thay đổi **connectString** xong, chúng ta tiến hành thử dùng chức năng đăng ký và thực hiện đăng ký một tài khoản thành viên mới.

Chạy lại project để đăng kí một user (/Account/Register) và xem kết quả trong database.

Application name Home About Contact

Register.
Create a new account.

Email

dinhnt.hutech@gmail.com

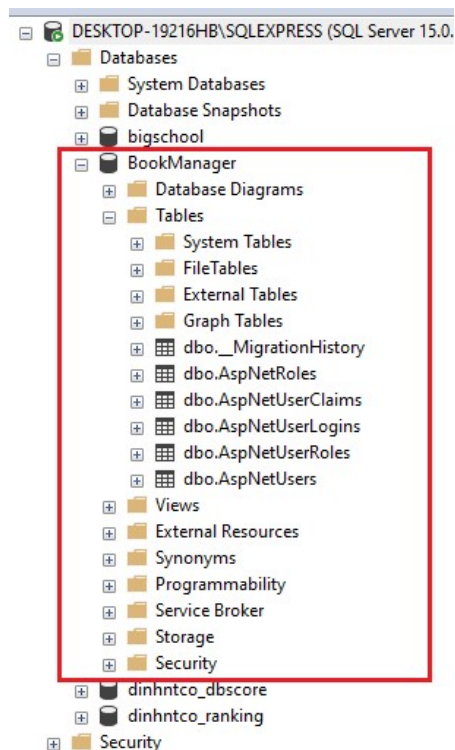
Password

Confirm password

Register

© 2021 - My ASP.NET Application

Sau khi đăng ký thành công, ta kiểm tra trong MS SQL Server. Database đã được tạo đúng như trong `connectString` đã khai báo.



Và tài khoản vừa đăng ký đã được lưu vào database

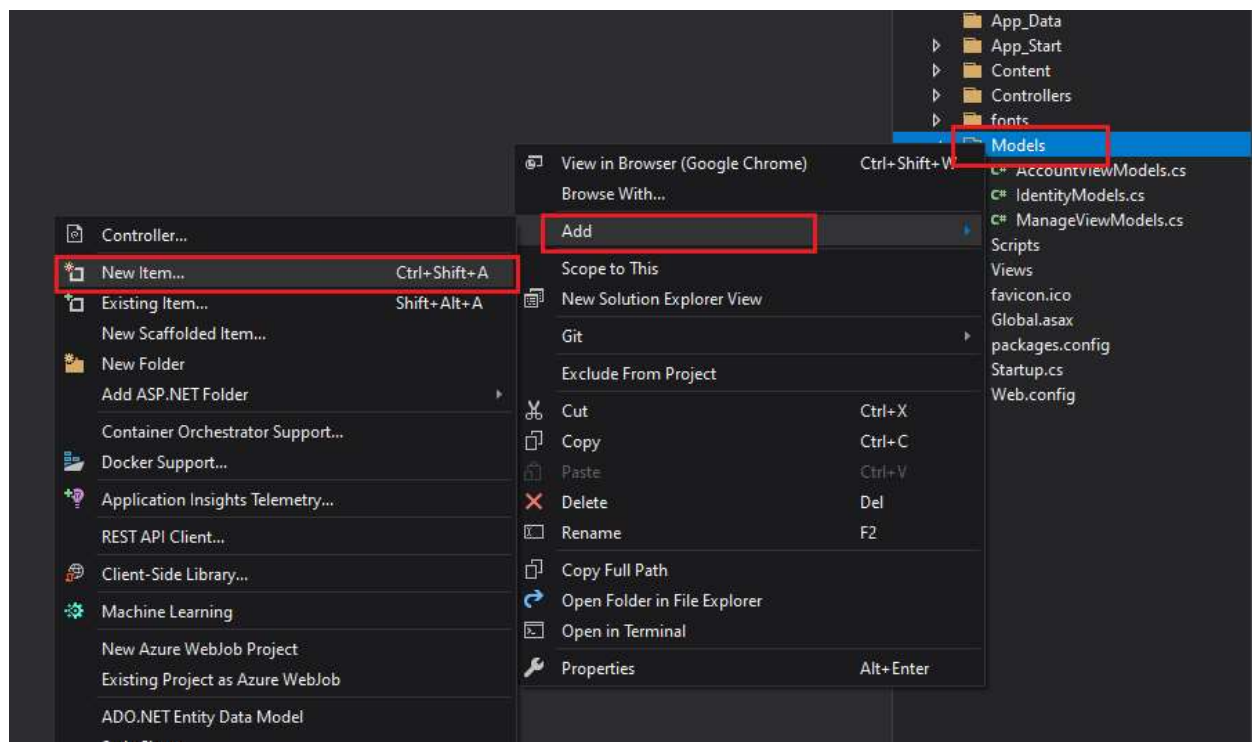
DESKTOP-19216HB\...- dbo.AspNetUsers								
	Id	Email	EmailConfirmed	PasswordHash	SecurityStamp	PhoneNumber	PhoneNumber...	
▶	af18-af480349a139	dinhnt.hutech@gmail.com	False	AlyR1G2zCOJbF...	53d09c94-71d5...	NULL	False	F
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	N

Bước 3: Thực hiện kết nối database vào project

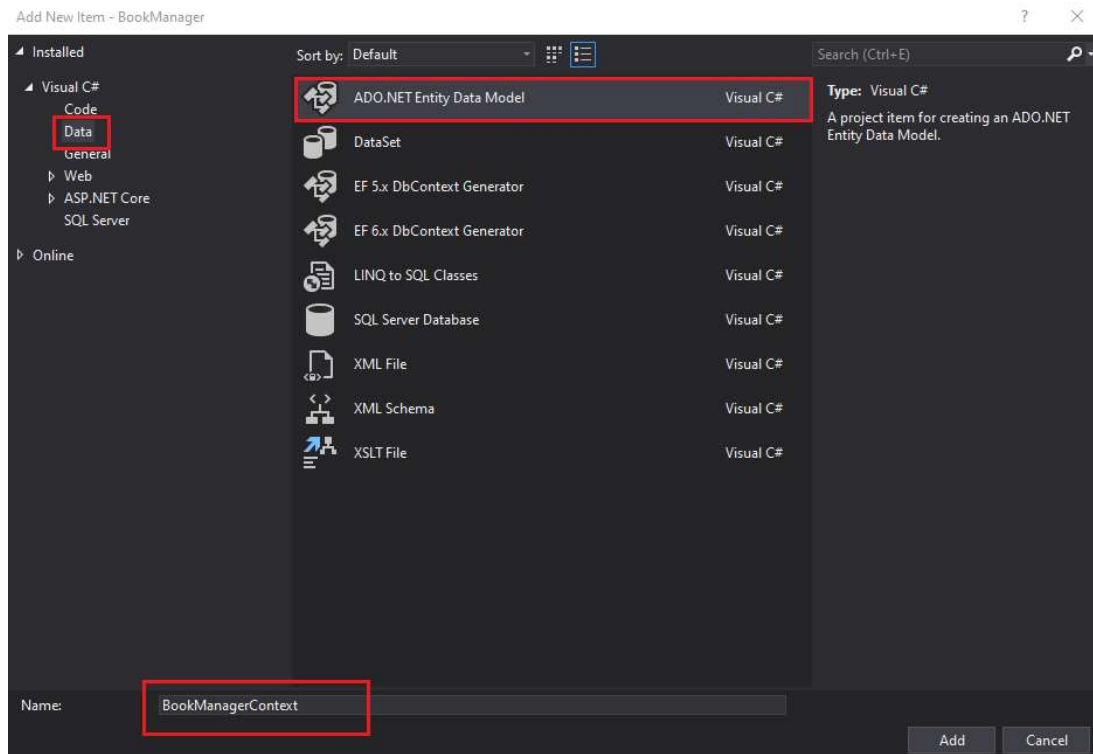
Thêm bảng **Book** vào database **BookManager** để quản lý thông tin của các cuốn sách, bao gồm các trường như sau:

DESKTOP-19216HB\...anager - dbo.Book			
	Column Name	Data Type	Allow Nulls
▶	ID	int	<input type="checkbox"/>
	Title	nvarchar(255)	<input type="checkbox"/>
	Description	nvarchar(255)	<input type="checkbox"/>
	Author	nvarchar(255)	<input type="checkbox"/>
	Images	nvarchar(255)	<input type="checkbox"/>
	Price	int	<input type="checkbox"/>

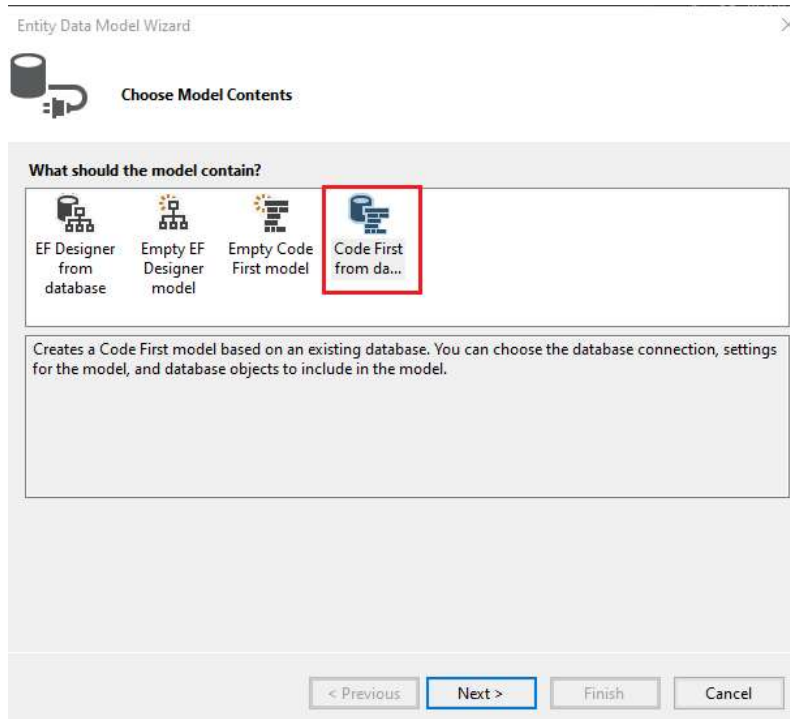
Thực hiện kết nối project với cơ sở dữ liệu (Sử dụng **Entity Framework Code First**). Click chuột phải vào folder **Models** → chọn **Add** → **New Item**



Chọn **Data** → chọn **ADO.NET Entity Data Model**
Đặt tên cho file là **BookManagerContext** → nhấn **Add**



Trong Entity Data Model Wizard, chọn **Code First from database** → nhấn **Next**



Chọn **connectString** (nếu đã có sẵn) hoặc tạo một **connectString** mới, đặt tên cho **connectString** đó và nhấn **Next**

Entity Data Model Wizard

Choose Your Data Connection

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

DefaultConnection (Settings) 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=DESKTOP-19216HB\SQLEXPRESS;initial catalog=BookManager;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework

☒ Save connection settings in Web.Config as:

BookContext

< Previous Next > Finish Cancel

Chọn table **Book** → nhấn **Finish**.

Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

☒ Tables

☒ dbo

☐ _MigrationHistory

☐ ASPNetRoles

☐ ASPNetUserClaims

☐ ASPNetUserLogins

☐ ASPNetUserRoles

☐ ASPNetUsers

☒ Book

☐ Views

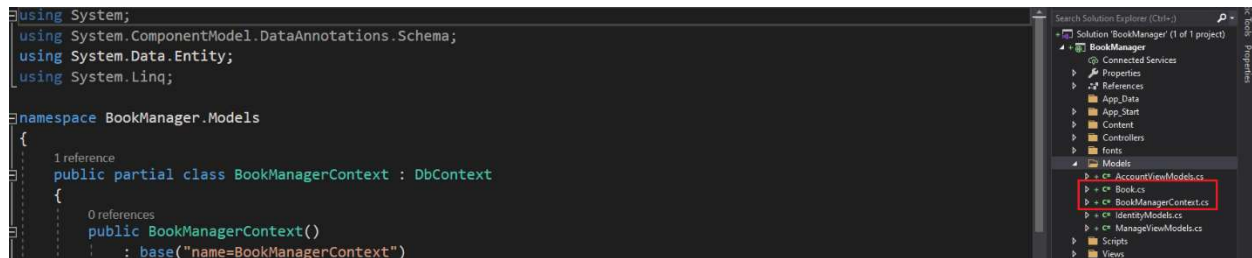
☒ Pluralize or singularize generated object names

☒ Include foreign key columns in the model

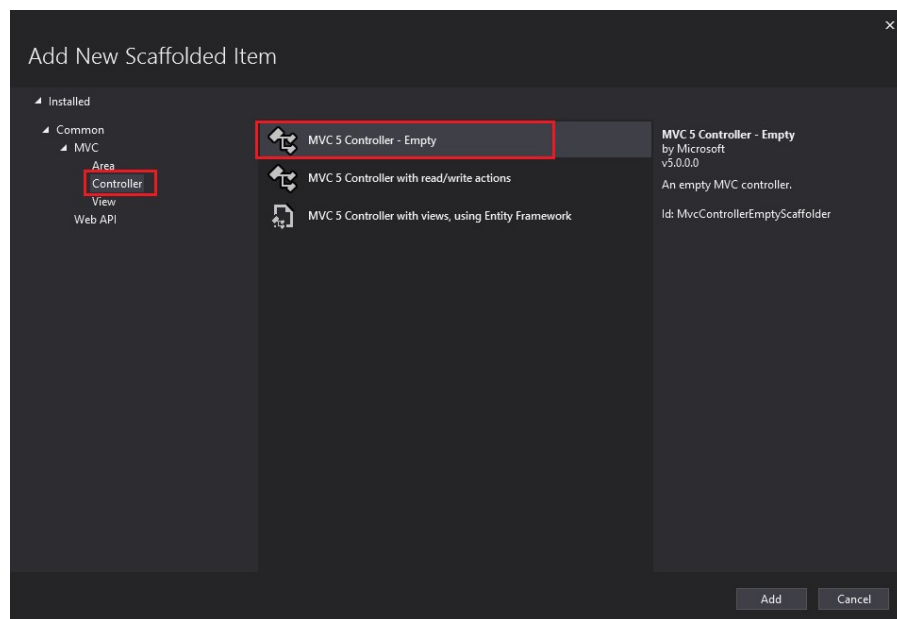
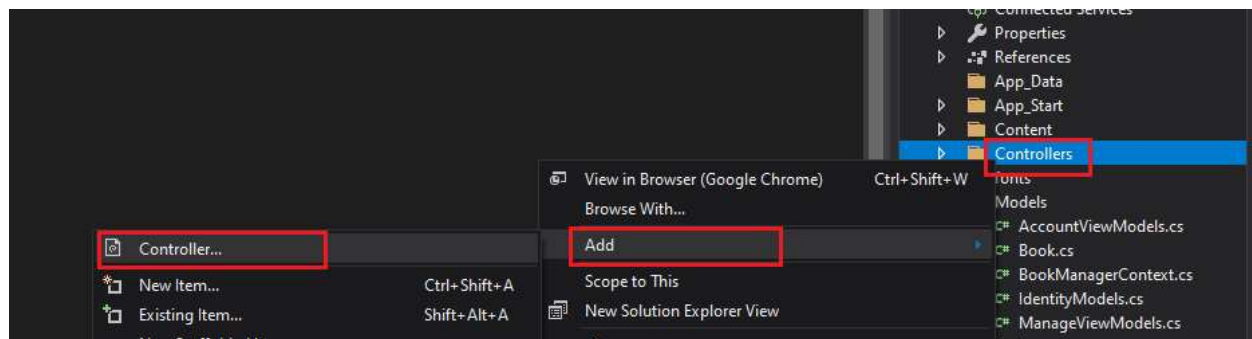
☐ Import selected stored procedures and functions into the entity model

< Previous Next > Finish Cancel

Sau khi kết nối CSDL xong, bảng **Book** trong CSDL được thêm vào **Models**



Nhấn chuột phải vào thư mục **Controllers** → Chọn **Add** → **Controller...** → Chọn **MVC5 Empty Controller**. Đặt tên cho controller là **BookController**.



Bước 4: Thực hiện chức năng hiển thị thông tin tất cả các cuốn sách có trong database

Viết một hàm mới có kiểu là **ActionResult** với tên hàm là **ListBook**. Trong hàm này ta sẽ thực hiện lấy thông tin của tất cả các cuốn sách ở bảng Book có trong database.

Sử dụng **Entity Framework Code First** để lấy thông tin tất cả các cuốn sách cho đối tượng **listBook** trong action **ListBook**

```
0 references
public class BookController : Controller
{
    0 references
    public ActionResult ListBook()
    {
        BookManagerContext context = new BookManagerContext();
        var listBook = context.Books.ToList();
        return View(listBook);
    }
}
```

Tạo view cho hàm ListBook bằng cách nhấn chuột phải vào tên hàm → Chọn **Add View** → chọn **MVC 5 View**. Trong cửa sổ **Add View**, lựa chọn **Template** là **List**, **Model Class** là **Book** và chọn layout cần sử dụng:

Add View

View name: ListBook

Template: List

Model class: Book (BookManager.Models)

Data context class:

Options:

☐ Create as a partial view

☒ Reference script libraries

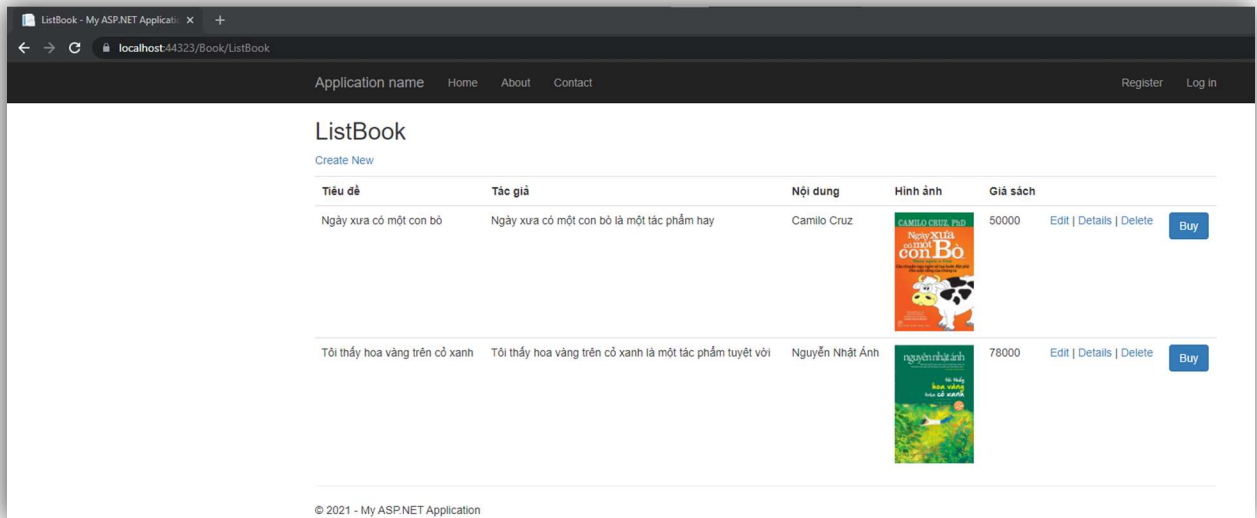
☒ Use a layout page:

~/Views/Shared/_Layout.cshtml

(Leave empty if it is set in a Razor _viewstart file)

Add Cancel

Sinh viên chỉnh sửa lại view **ListBook** cho phù hợp và *tự nhập một số dòng dữ liệu* vào database. Sau đó chạy chương trình để được kết quả như hình:



Bước 5: Thực hiện chức năng mua sách

Viết một hàm **Buy** trong **BookController** để thực hiện thao tác mua sách.

```
public ActionResult Buy(int id)
{
    BookManagerContext context = new BookManagerContext();
    Book book = context.Books.SingleOrDefault(p => p.ID == id);
    if(book == null)
    {
        return HttpNotFound();
    }

    return View(book);
}
```

Chúng ta thêm **[Authorize]** trên hàm **Buy** trong **BookController** để bắt buộc người dùng phải đăng nhập trước khi thực hiện mua sách.

```
[Authorize]
// references
public ActionResult Buy(int id)
{
    BookManagerContext context = ne
```

Tạo view cho hàm **Buy**

```
@model BookManager.Models.Book

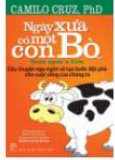

@{
    ViewBag.Title = "Buy";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<br/>
<p style="font-size:24px">Bạn đã mua cuốn sách <strong>@Model.Title</strong> với giá <strong>@Model.Price</strong> VNĐ</p>
```

Trong view **ListBook** thêm một cột trong table danh sách để hiển thị nút **Buy** cho từng cuốn sách:

```
</td>
<td>
    @Html.ActionLink("Edit", "Edit", new { id=item.ID }) |
    @Html.ActionLink("Details", "Details", new { id=item.ID }) |
    @Html.ActionLink("Delete", "Delete", new { id=item.ID })
</td>
<td>
    <a href="@Url.Action("Buy","Book", new { id = item.ID })" class="btn btn-primary" >Buy</a>
</td>
</tr>
```

Chạy lại chương trình và xem kết quả

ListBook						
Create New						
Tiêu đề	Nội dung	Tác giả	Hình ảnh	Giá sách		
Ngày xưa có một con bò	Ngày xưa có một con bò là một tác phẩm hay	Camilo Cruz		50000	Edit Details Delete	Buy
Tôi thấy hoa vàng trên cỏ xanh	Tôi thấy hoa vàng trên cỏ xanh là một tác phẩm tuyệt vời	Nguyễn Nhật Ánh		78000	Edit Details Delete	Buy

Khi click vào nút **Buy**, người dùng bắt buộc thực hiện đăng nhập nếu như chưa đăng nhập

Log in - My ASP.NET Application

localhost:44323/Account/Login?ReturnUrl=%2FBook%2FBuy%2F2

Application name Home About Contact

Log in.

Use a local account to log in.

Email

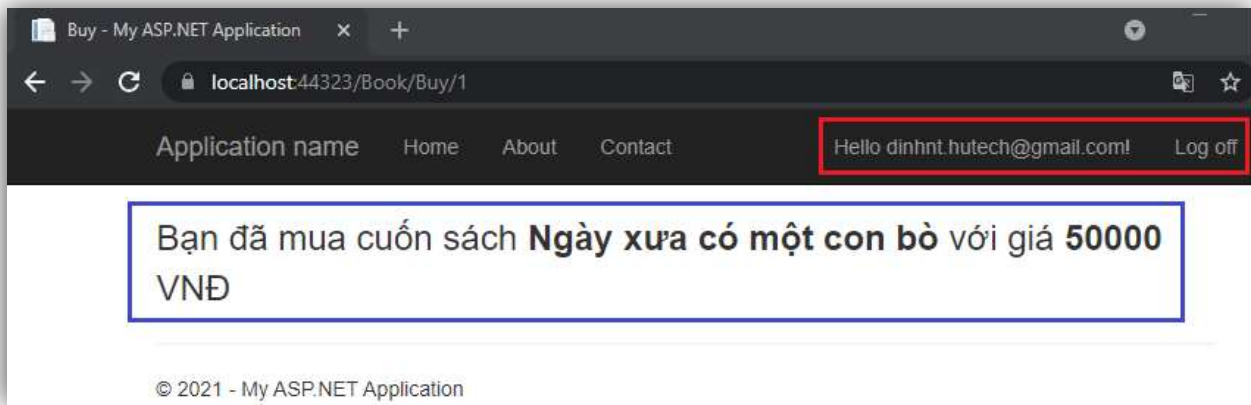
Password

☐ Remember me?

[Register as a new user](#)

© 2021 - My ASP.NET Application

Giao diện sau khi đăng nhập và người dùng thực hiện mua sách



Bước 6: Sinh viên thực hiện các chức năng còn lại (Create, Edit, Delete)

Hướng dẫn: Sử dụng Entity để lấy danh sách, thêm , xóa , sửa với CSDL

```
//Sử dụng context để làm việc với các class
BookManagerContext context = new BookManagerContext();

//1. Lấy tất cả thông tin sách từ bảng Book
List<Book> listBook = context.Books.ToList();

//2. Lấy thông tin một cuốn sách trong bảng Book khi có BookID cho trước
Book db = context.Books.FirstOrDefault(p => p.ID == BookID);

//3. Insert một đối tượng Book b vào database
context.Books.AddOrUpdate(b); //Add or Update Book b
context.SaveChanges();

//4. Update thông tin sách vào database
Book dbUpdate = context.Books.FirstOrDefault(p => p.ID == BookID);
if(dbUpdate != null){
    context.Books.AddOrUpdate(b); //Add or Update Book b
    context.SaveChanges();
}

//5. Xóa Book khi có BookID cho trước, tương tự update
Book dbDelete = context.Books.FirstOrDefault(p => p.ID == BookID);
if (dbDelete != null) {
    context.Books.Remove(db);
    context.SaveChanges();
}
```