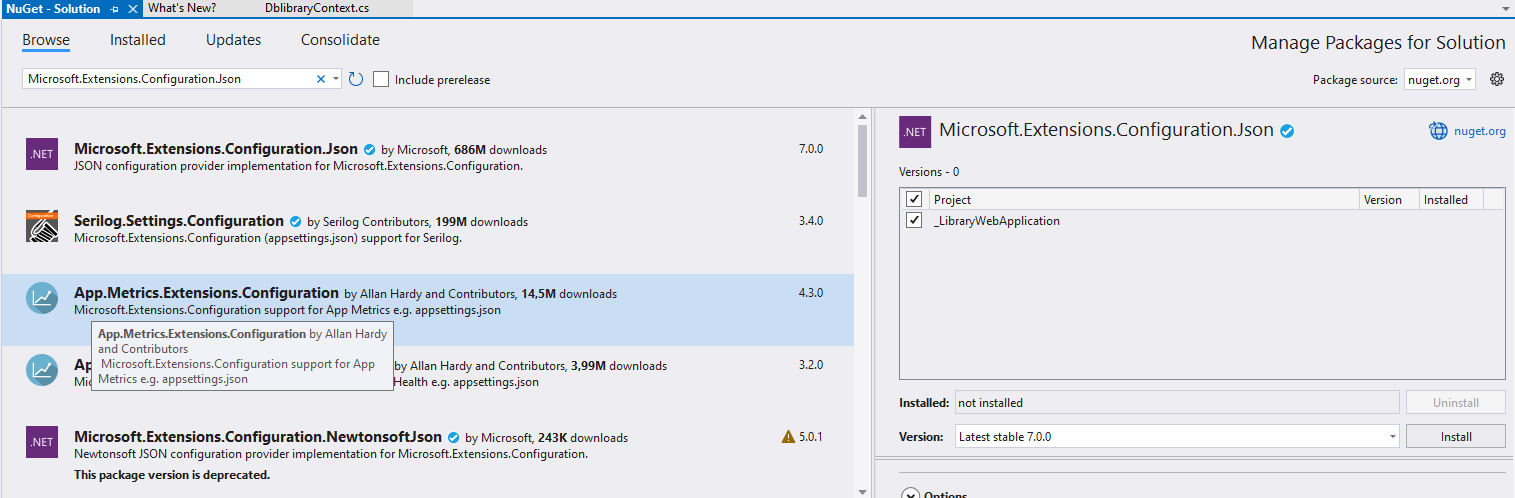
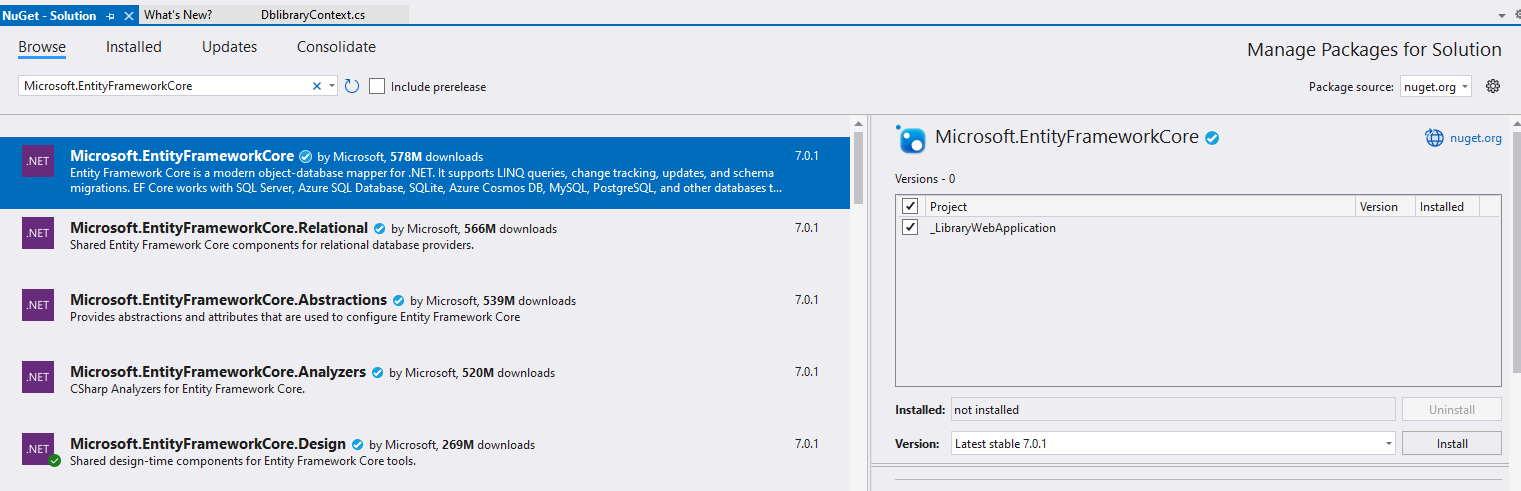
**ЛАБОРАТОРНА РОБОТА №2. ЕТАП 3**

ТУТ НАВЕДЕНО ЛИШЕ ІЛЮСТРАТИВНІ ПРИКЛАДИ. ПРИ РЕАЛІЗАЦІЇ СВОГО ПРОЄКТУ ВИ ПОВИННІ РЕАЛІЗУВАТИ **ПОВНОФУНКЦІОНАЛЬНИЙ** ЗАСТОСУНОК!

1. Для роботи з конфігурацією в форматі json додамо в проєкт Nuget пакет Microsoft.Extensions.Configuration.Json:

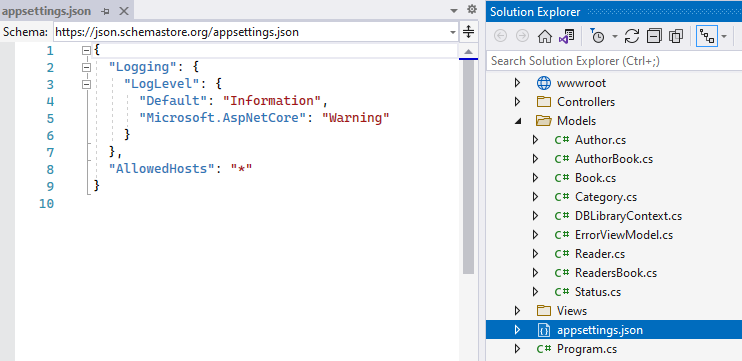
****

Для роботи з EF - Microsoft.EntityFrameworkCore

****

1. **Рядок підключення**

Для підключення до бази даних, нам треба задати параметри підключення. Для цього змінимо файл appsettings.json. За замовчуванням він містить тільки налаштування логування:

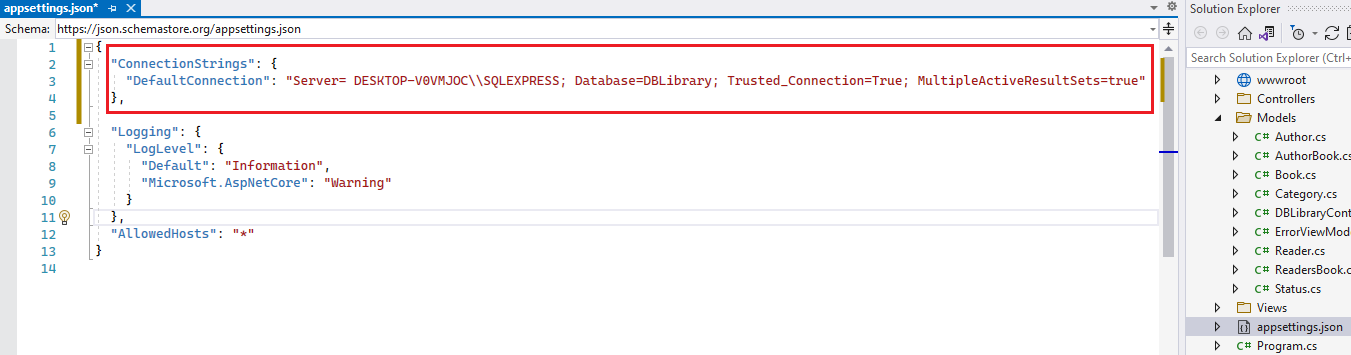


Внесемо зміни, додавши рядки підключення (***параметри рядка підключення див. в ЛР2\_Етап\_2\_ІСтаТП.docx***):

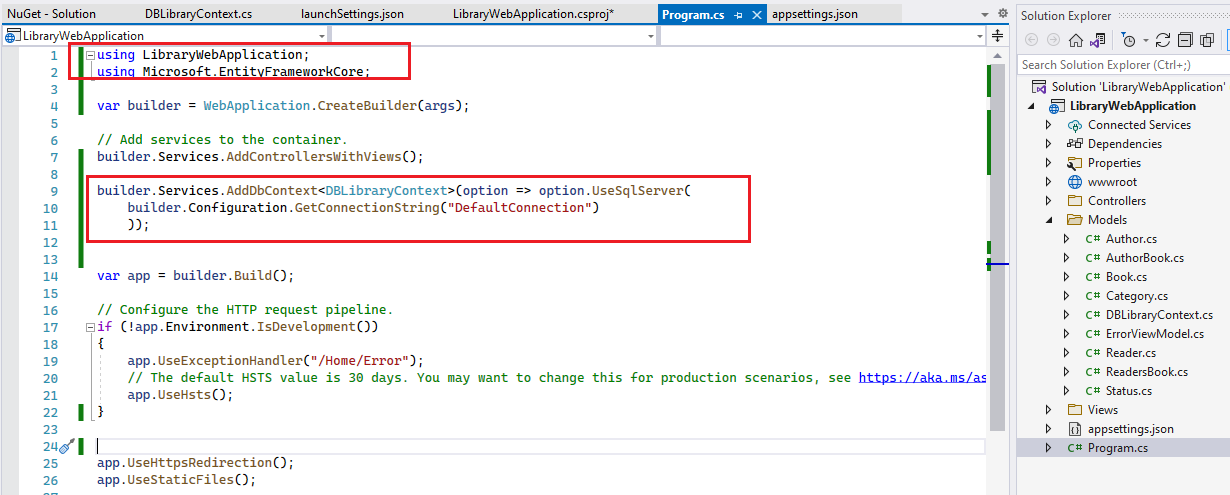
"ConnectionStrings": {

"DefaultConnection": "Server= *DESKTOP-V0VMJOC****\\****SQLEXPRESS*; Database=*DBLibrary*; Trusted\_Connection=True; MultipleActiveResultSets=true"

},



1. **Зміни в файлі Program.cs**



builder.Services.AddDbContext<**DBLibraryContext**>(option => option.UseSqlServer(

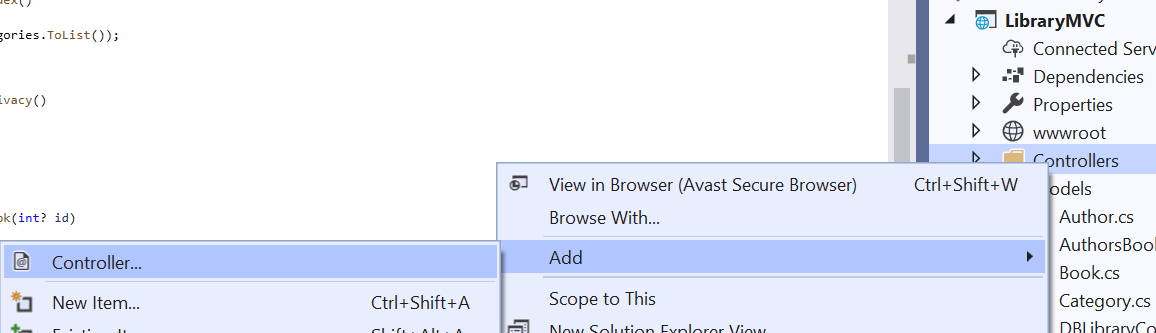
builder.Configuration.GetConnectionString("DefaultConnection")

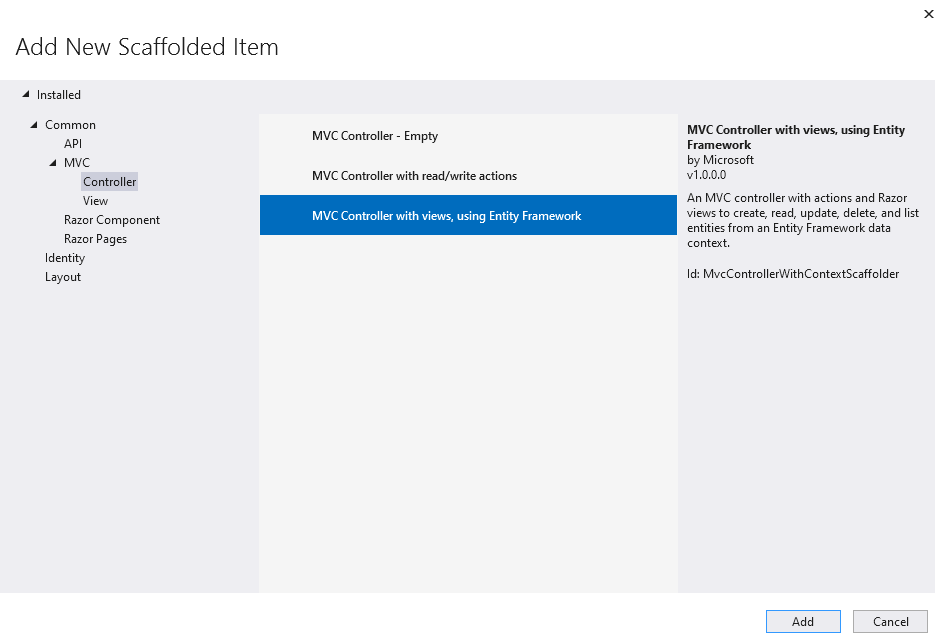
));

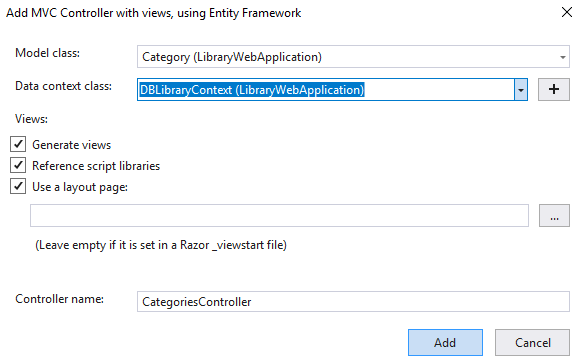
1. **Створення контролера для Entity Framework**

Для виведення переліку категорій книжок на веб-сторінку, щоб користувачі змогли обрати одну з них потрібно створити метод (дію контролера) і представлення, яке відповідатиме за відображення потрібної інформації. Продемонструємо створення контролера для класу моделі Entity Framework.

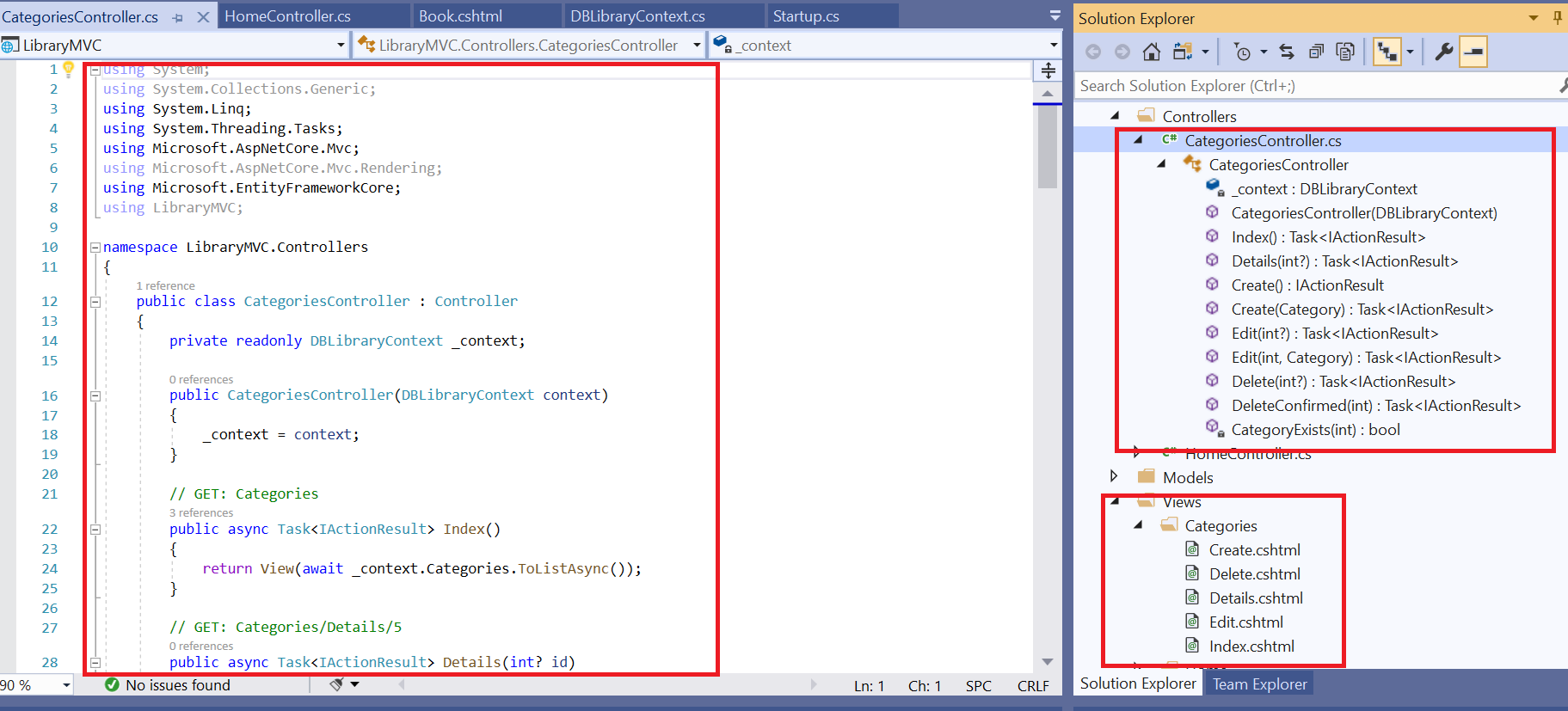
Для цього перейдемо до папки Controllers, де поки що знаходиться єдиний контролер HomeController.На папці Controllers натиснемо праву кнопку миші та виберемо:



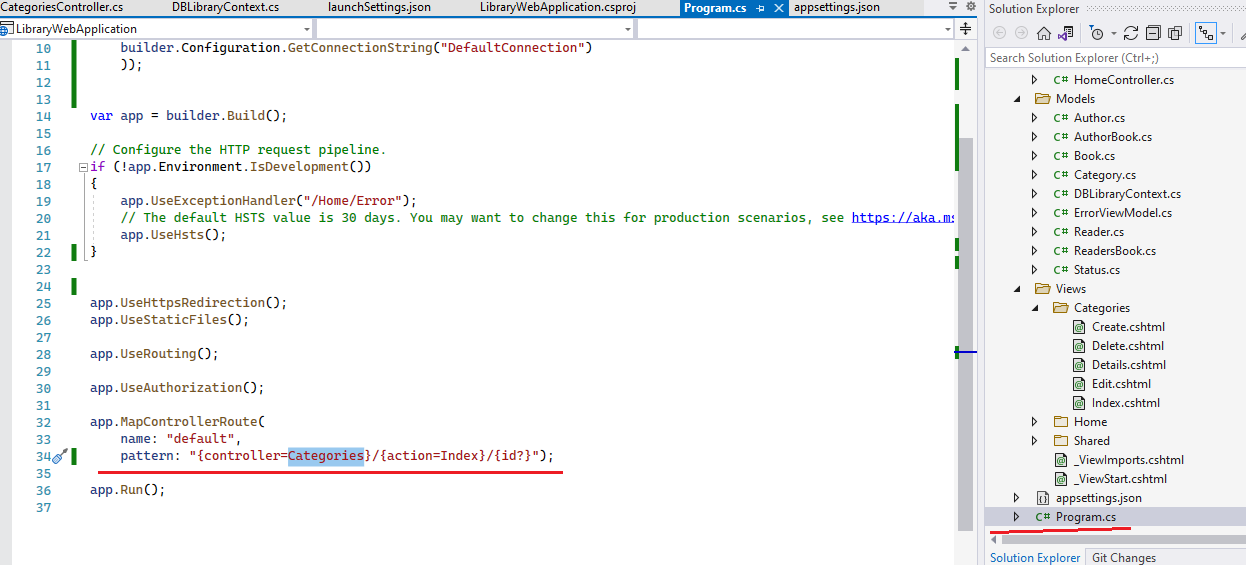


В налаштуваннях виберемо потрібний клас моделі та клас контексту даних. Поле з layout ПОКИ що (до наступного етапу) залишаємо порожнім:

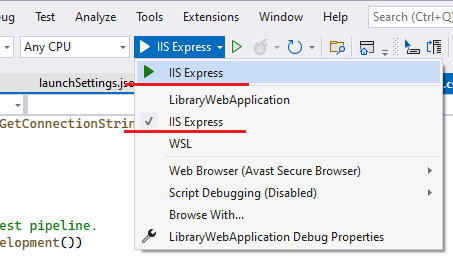
Зверніть увагу, що щойно було згенеровано не лише файл контролера, а і низка файлів представлення. Перегляньте файли, що згенерувалися (поки що без внесення змін):



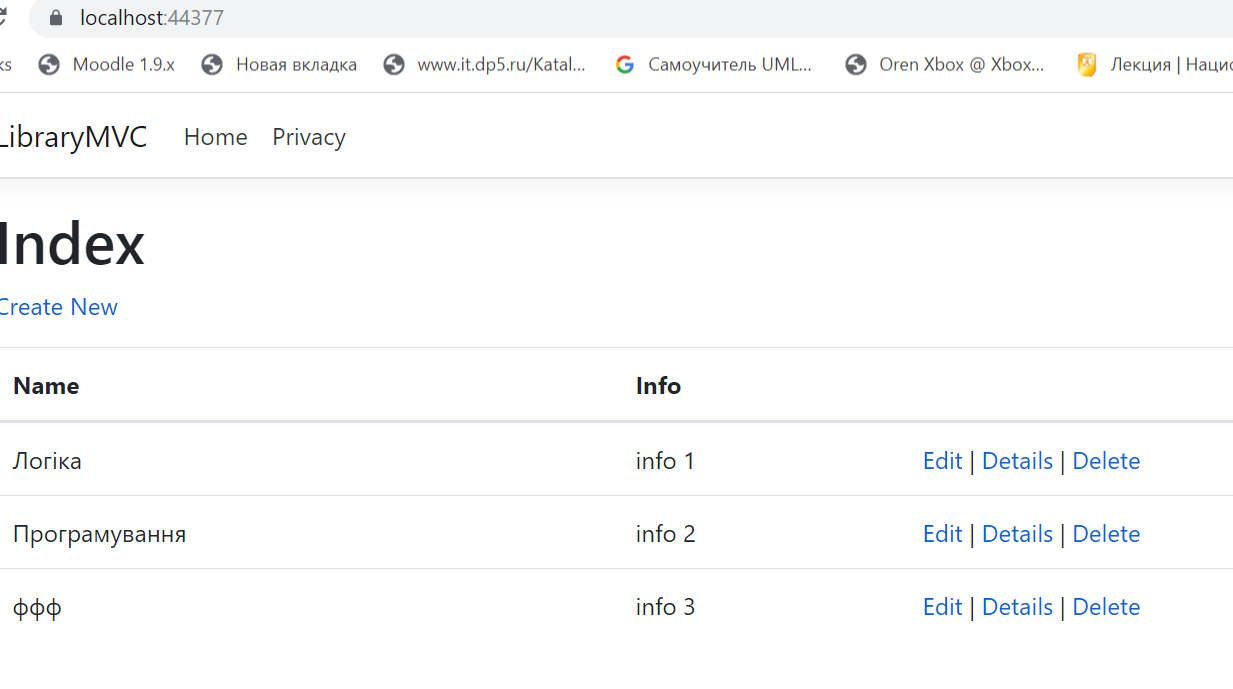
Проте, якщо зараз запустити додаток, то ці сторінки не будуть запущені за замовчуванням. Для того, щоб зробити саме сторінку Index контролера Categories такою, що запускається по замовчуванню внесемо зміни у файл Program.cs (в корені проєкту):

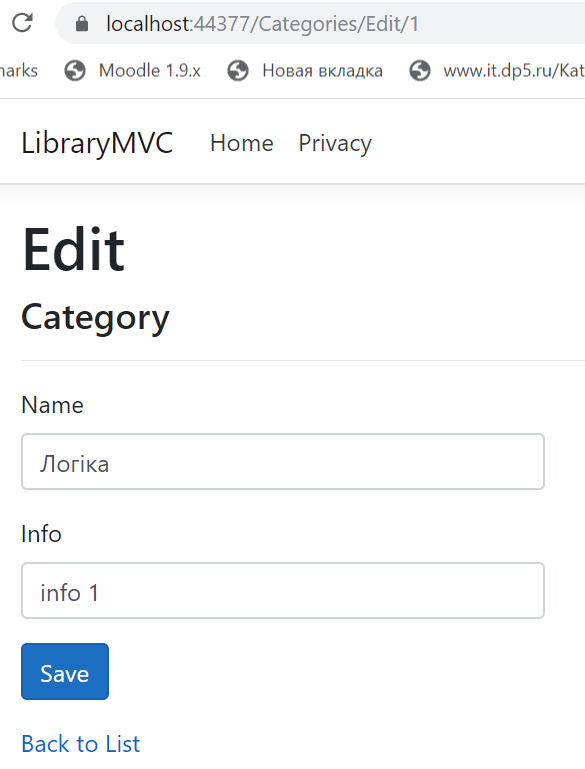


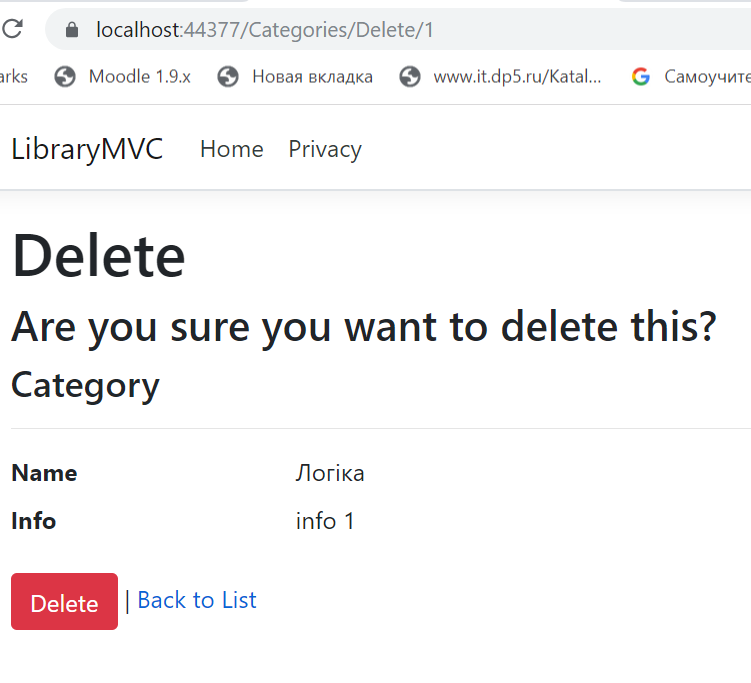
Запустимо проєкт.



Неважко переконатися, що ми отримала достатньо багато реалізованого функціоналу:





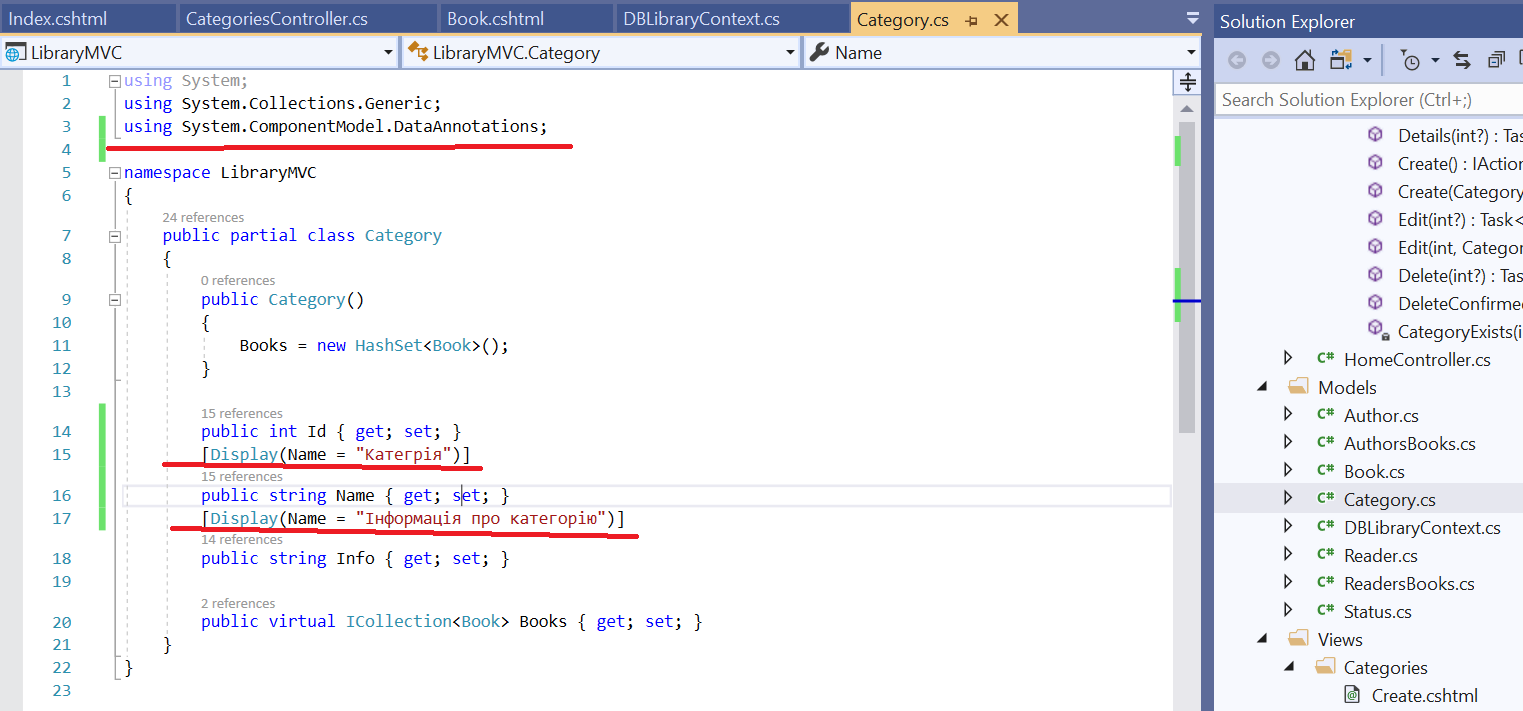
****

Проте, не все нам подобається. Зверніть увагу на назви, які відображаються, вони нам зовсім не подобаються ☹

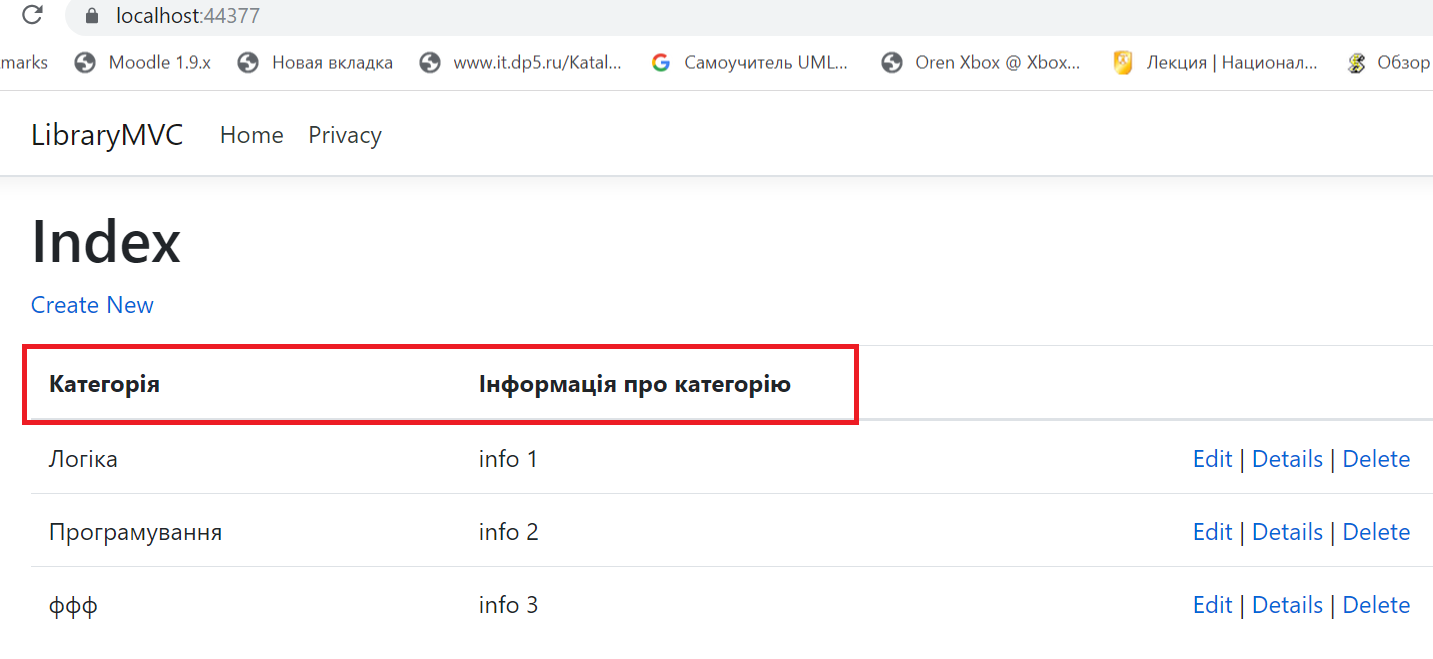
Переглянемо (поки не змінюємо) файл Index.cshtml (в папці Views/Categories):



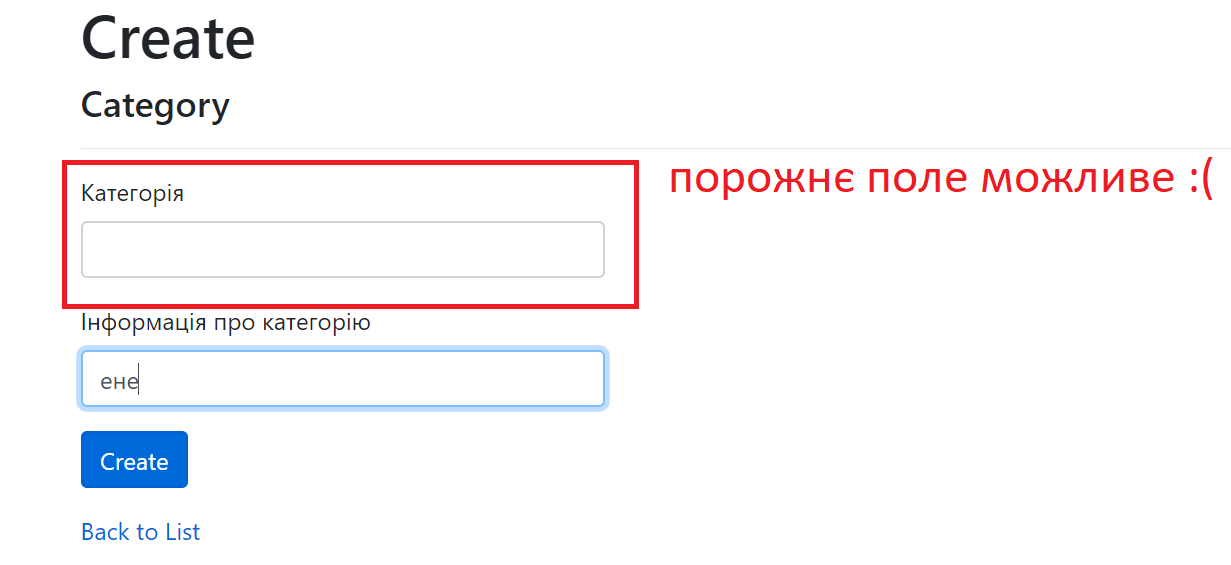
Як видно, ці назви в явному вигляді не виводяться. Проблема в моделі. Внесемо зміни (додамо атрибути) у файл з моделлю Category:



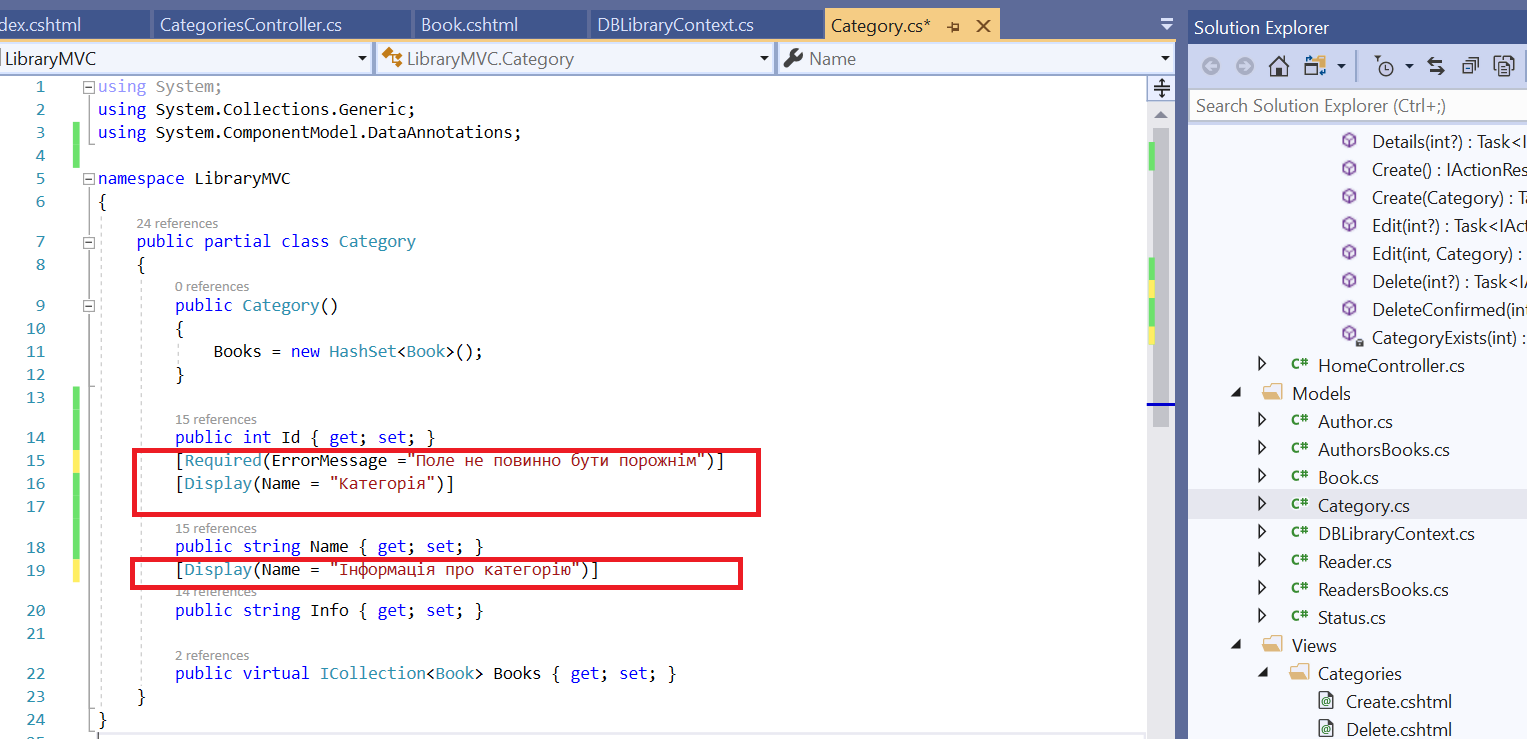
Запускаємо ☺:



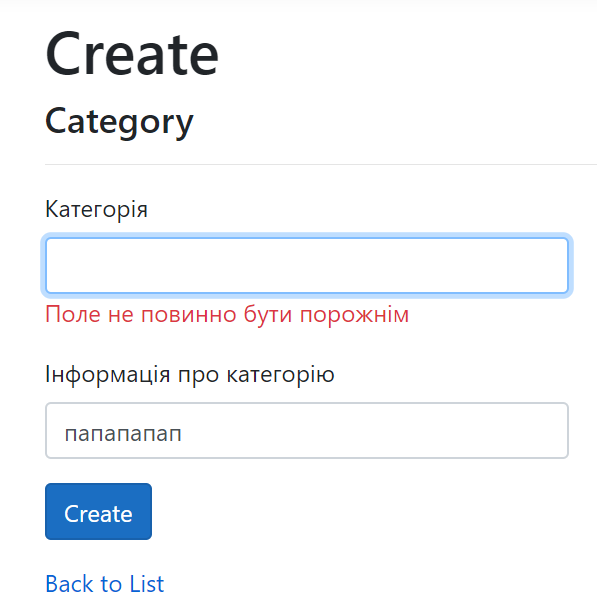
Але, ми і досі не задоволені. Зокрема, при створенні нової категорії поле з іменем можна залишити порожнім, але при збереженні в БД виникає помилка, адже там порожнє ім’я заборонене:



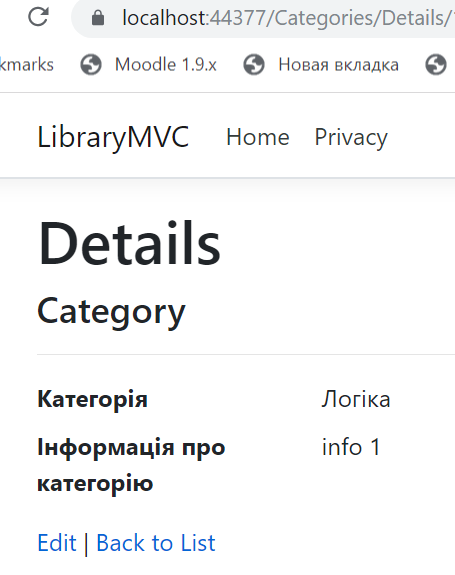
Для валідації (перевірки коректності) даних знову скористаємося додаванням атрибутів в модель (з більшою кількістю таких атрибутів ознайомтеся в лекції):



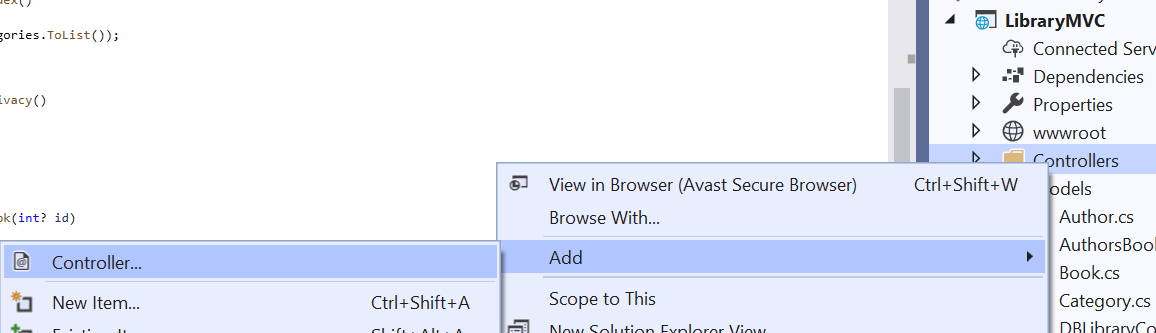
Запускаємо і радіємо ☺:

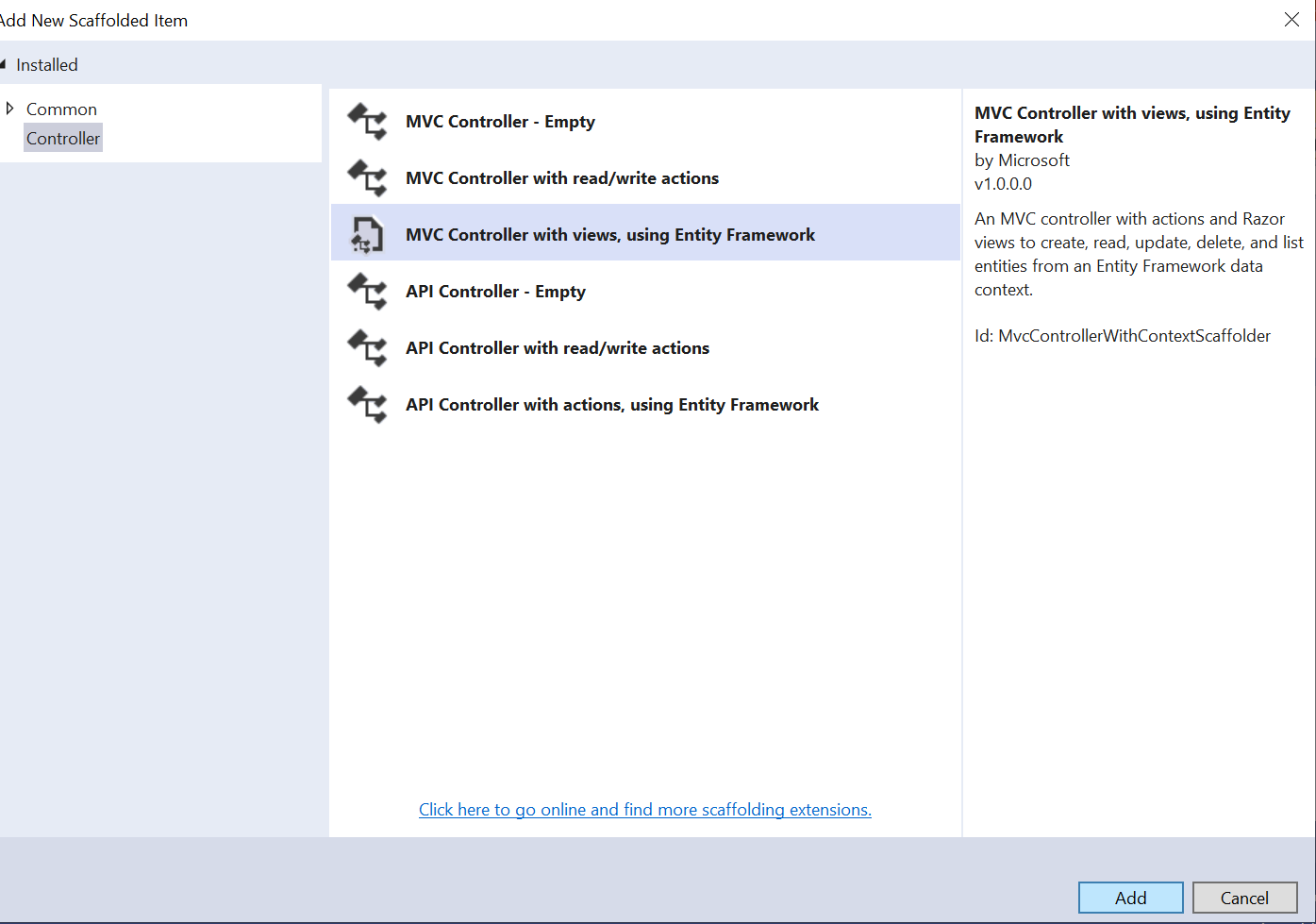
****

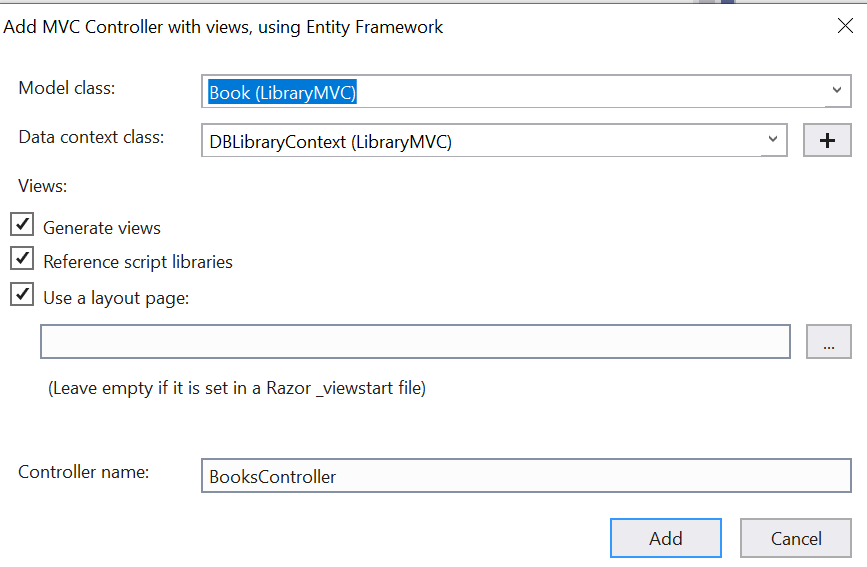
Тепер, нам не подобається вікно з детальною інформацією про категорію книг (Details). На ньому, крім інформації про назву категорії бажано було б мати і перелік книг з цієї категорії з можливістю їх додавання перегляду та вилучення:

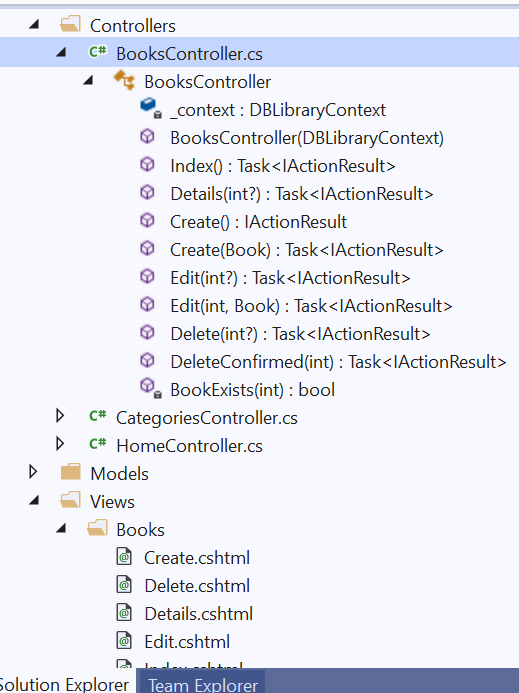


Створимо контролер для класу моделі Book (аналогічно до Categories):





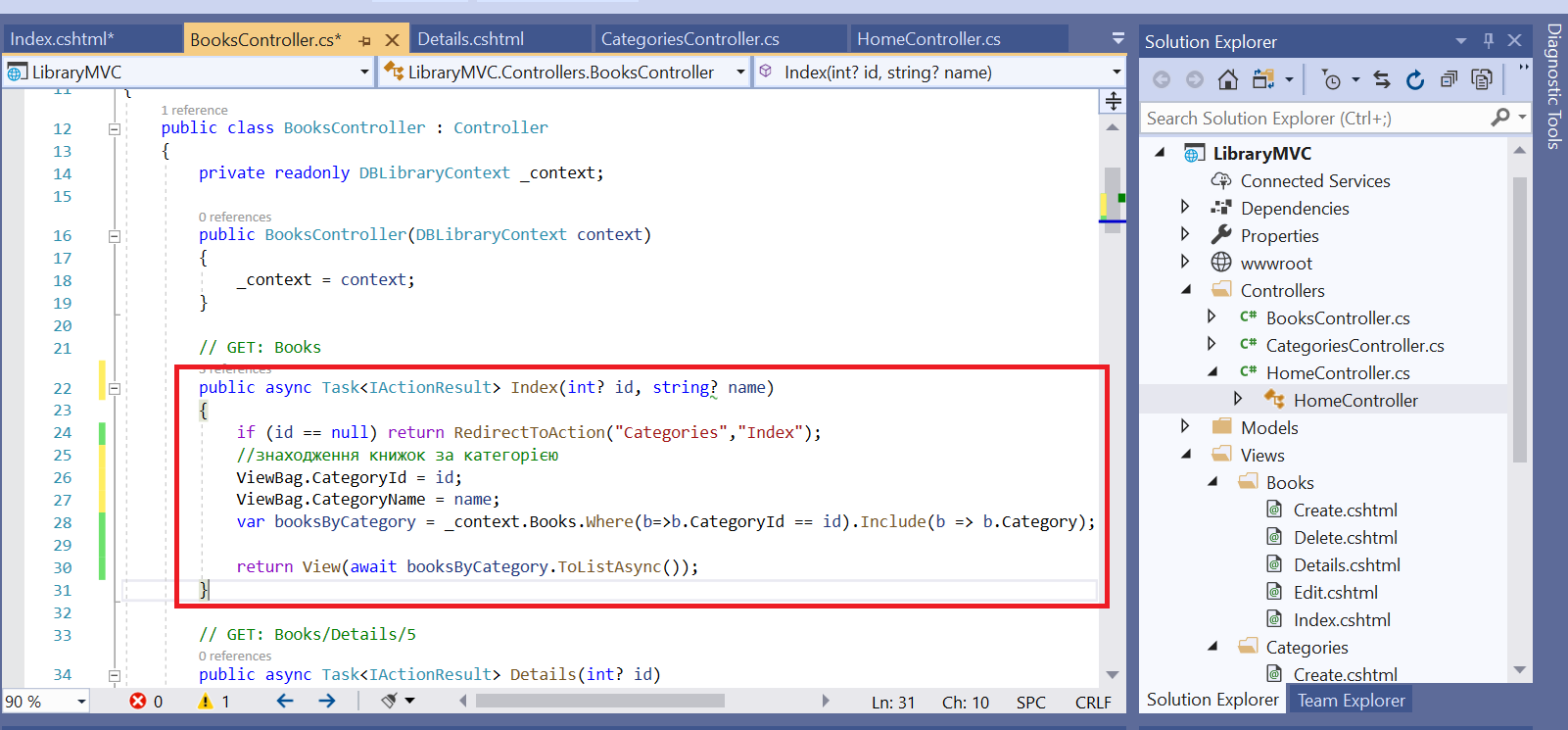




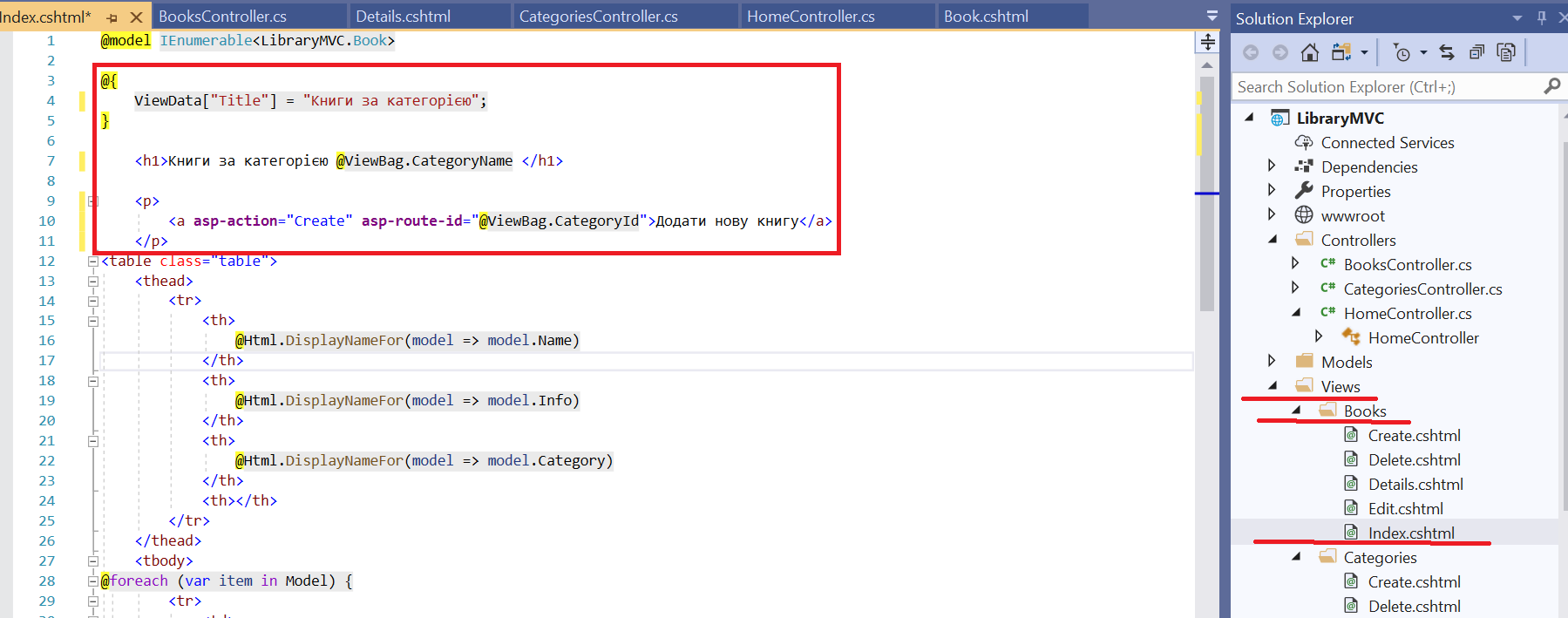
Для того, щоб при натисканні на посилання «Details» з інформацією про потрібну категорію ми переходили на відображення книг за цією категорією внесемо зміни в контролер категорій:



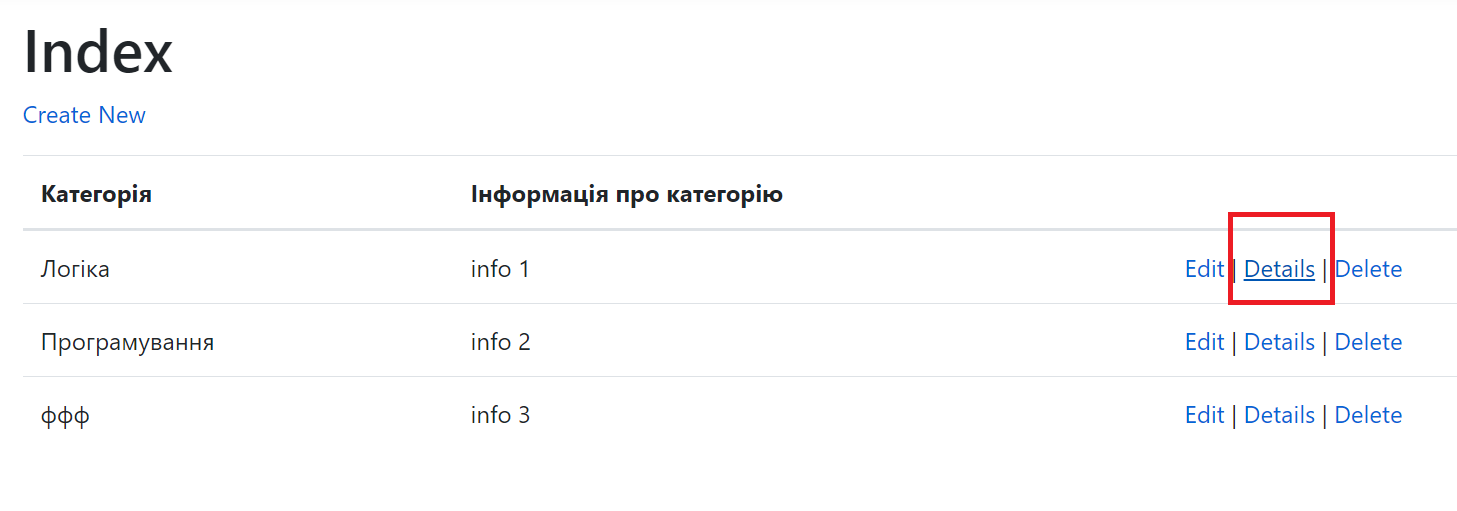
Тепер «навчимо» метод Index контролера «Books» розуміти інформацію про категорію і обробляти її. Для цього внесемо зміни у відповідний файл:

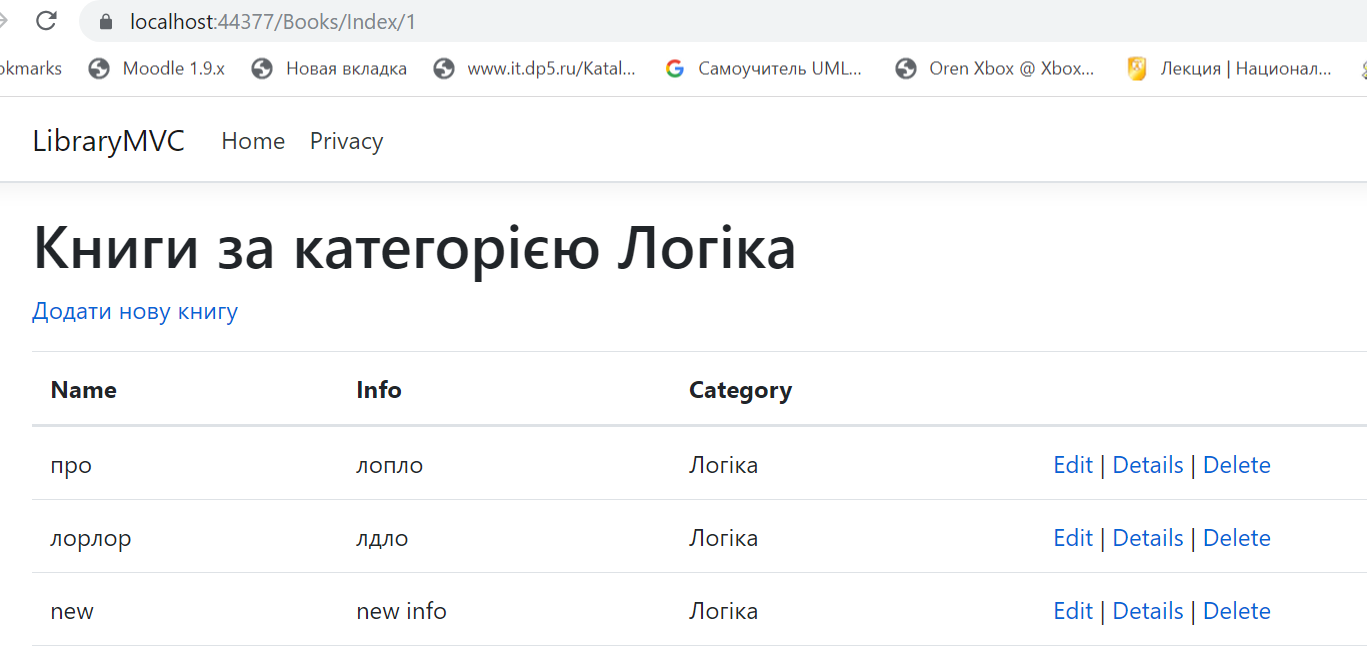


Для коректного відображення сторінки з книжками внесемо зміни і у файл відображення книжок “Index”:



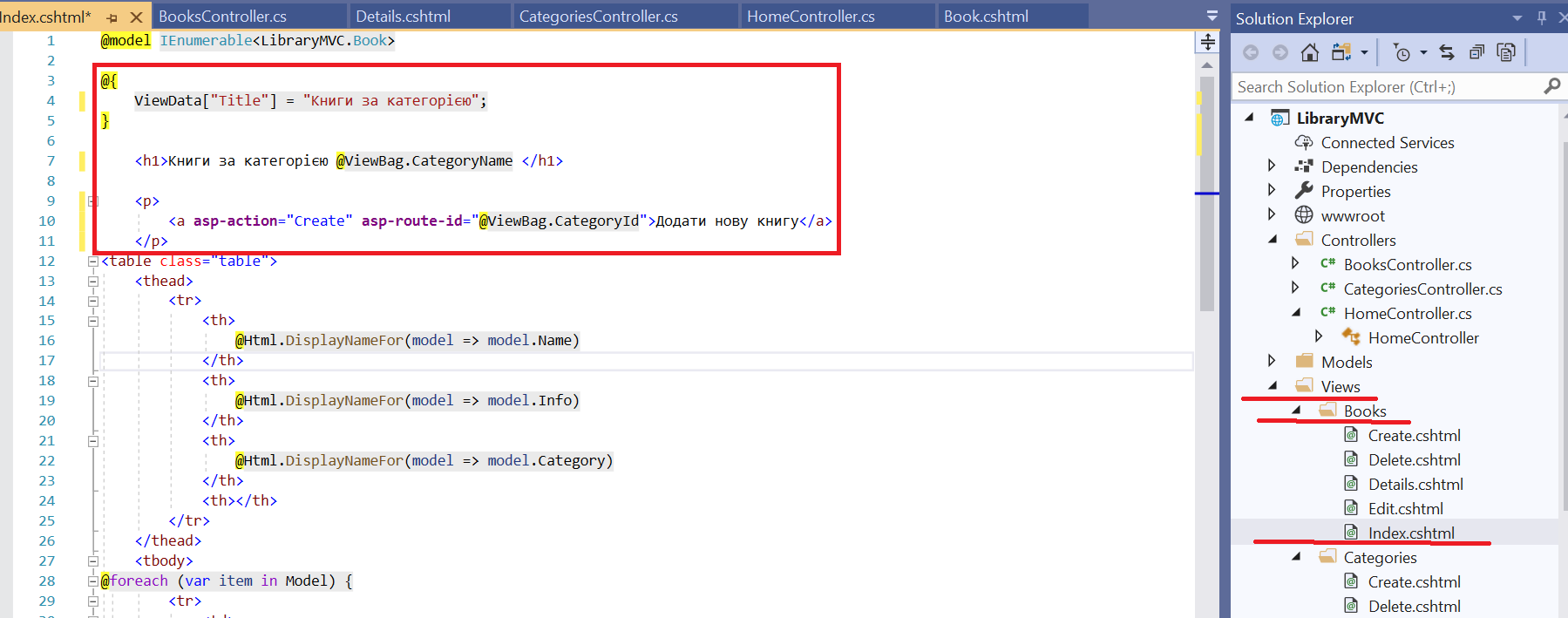
Можна запускати ☺:



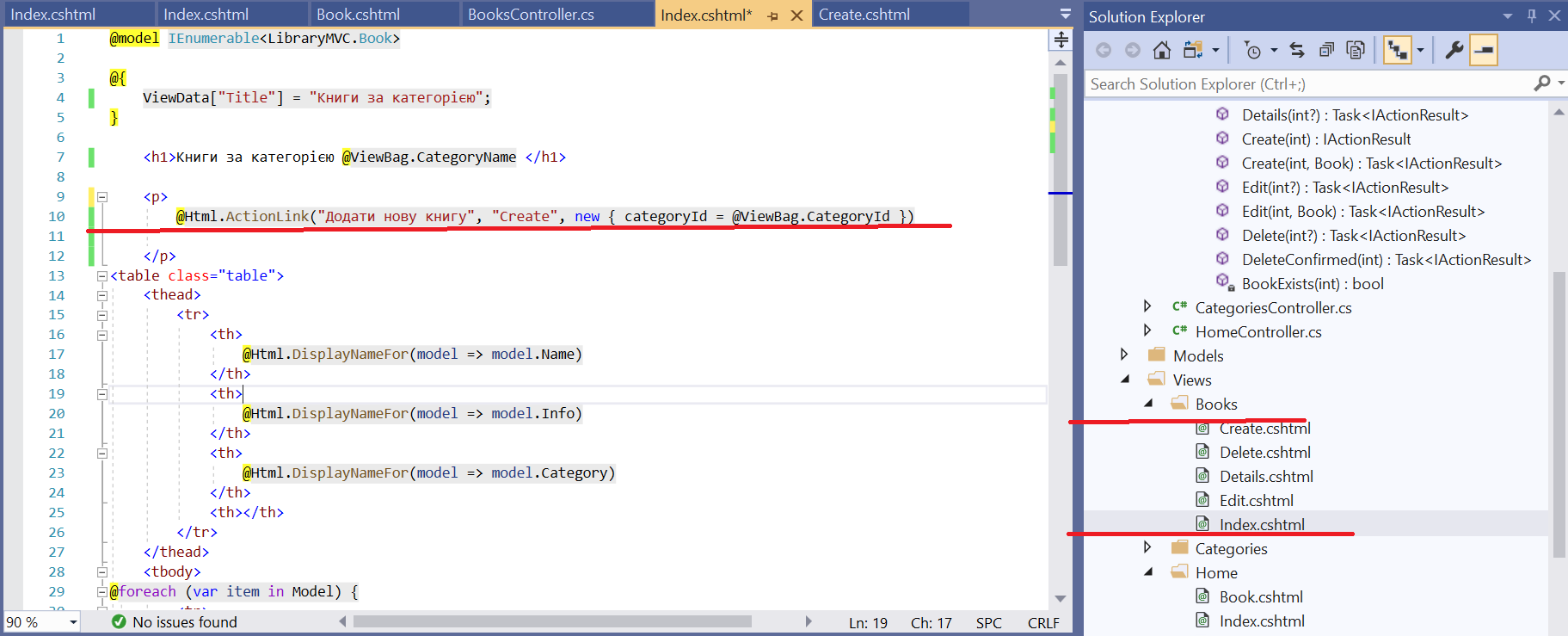


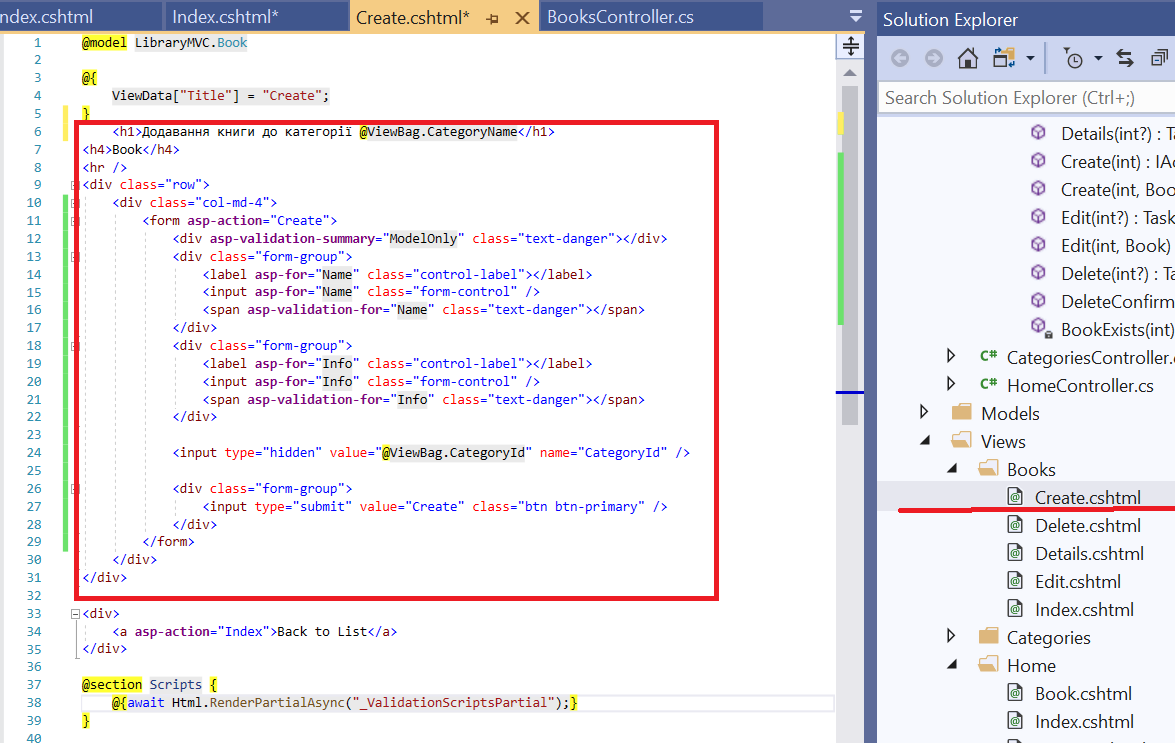
З правильними підписами та валідацією даних розберетеся самостійно ☺.

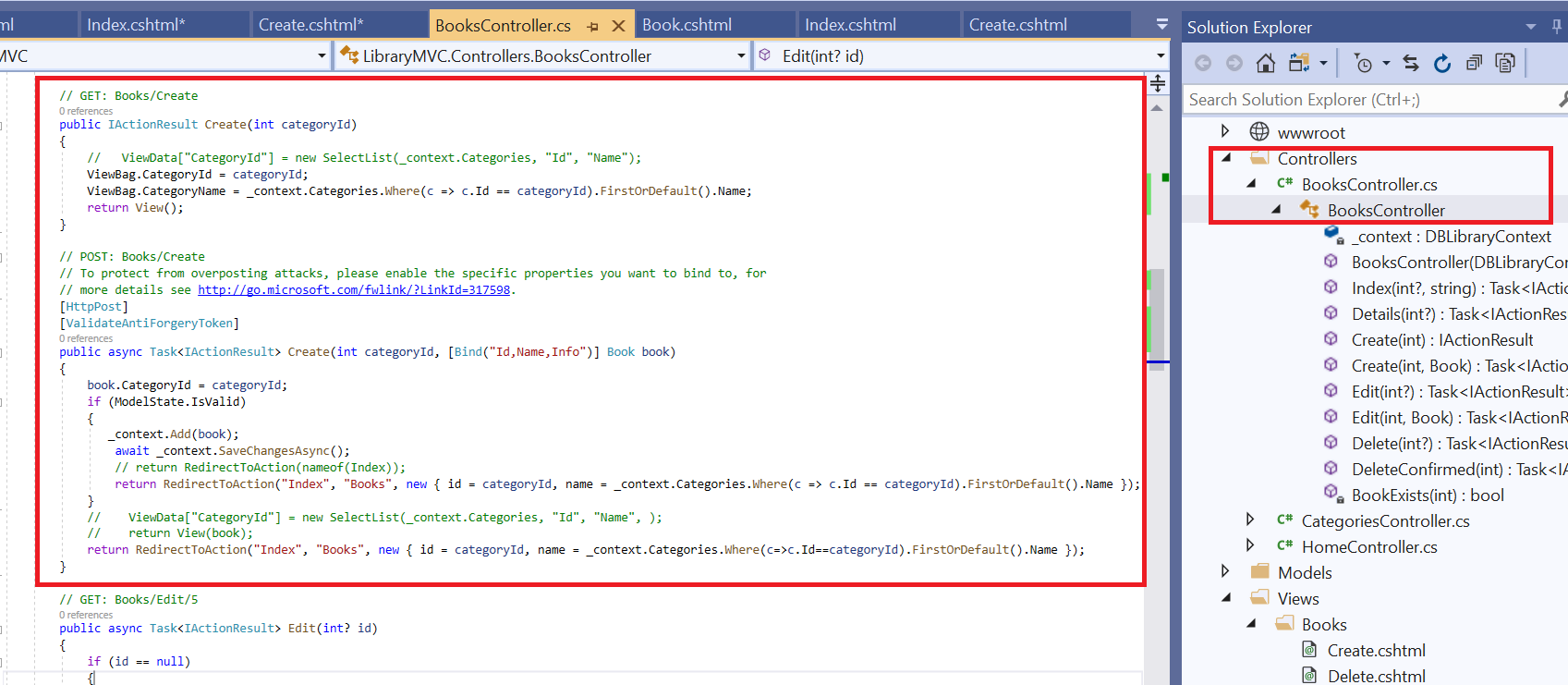
Розберемося з додаванням нової книги в категорію, а саме опрацюємо посилання «Create», яке ми попередньо додали у представлення (рядок 10):



Для цього потрібно внести зміни у відповідний метод контролера Books та у відповідне представлення. Тепер цей метод повинен приймати у якості параметра ідентифікатор категорії:







Можна запускати ☺.

ДАЛІ САМОСТІЙНО!

Для отримання балів за етап – реалізуйте **повнофункціональний** застосунок.

Викладач має право критикувати ☺.

Складність залежить від обраної Вами предметної області.

Для .NET 6

Якщо для зв’язаних сутностей не додаються записи, то

Почитайте тут: <https://stackoverflow.com/questions/70415290/asp-net-core-6-mvc-ef-core-6-model-is-not-validating-correctly>

А саме:

Beginning with .NET 6, new projects include the <Nullable>enable</Nullable> element in the project file. Once the feature is turned on, existing reference variable declarations become **non-nullable reference types**.

Тому, знайдіть в проєкті рядок <Nullable>enable</Nullable> та видальте.

