# Exercises: Basics. Razor Pages

Problems for exercises and homework for the [“C# MVC Frameworks - ASP.NET” course @ SoftUni](https://softuni.bg/courses/asp-net-mvc).

You have an enormous book library. You want to keep track of all books and you want to know where each one is placed.

Being a generous person and a good friend, you often give books to your friends to borrow. However, you’ve noticed that most people don’t return the books on time (or at all).

Write an application which stores information about your books and who keeps them. Use ASP.NET Core pages for every exercise.

**Use Bootstrap 4 for the styles. The ASP.NET Core template uses 3.3.7 by default.**

## Data Models

* **Book**: title, short description, author, book cover image (optional)
* **Author**: name, books
  + I recommend that you create this as another table in order to make searching for authors quick
  + Assume one author per book. If there are many authors, simply save their names together
* **Borrower**: name, address

You will also need other models in order to create the app but you’ve got to figure these on your own.

Feel free to add other models and extend the current ones as you wish.

## Database and Data Layer

Use **Entity Framework Core**. You can create your database models and context in your Web app project but it’s better to separate them in other projects (class libraries).

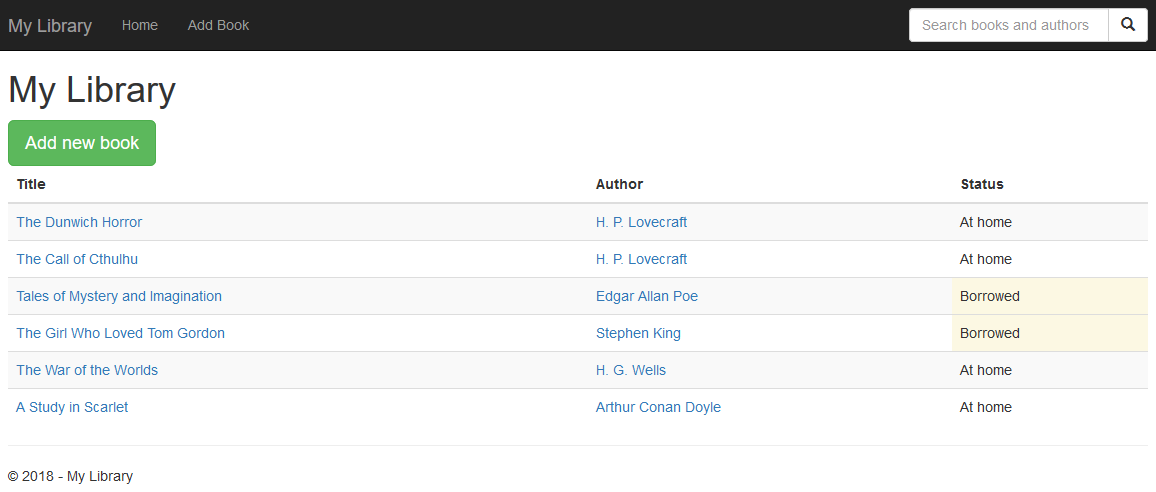
Add validation to the EF Core models. You don’t need to bother validating the forms for now.

## Display All Books

At the front page (**/**) you’ll get a list of all existing books in the database, and whether they have been borrowed or not.

The book title takes you to the book details page. The author name takes you to the author details page.

Don’t worry about the **Status** column for now. We’re going to manage that in the next exercise.



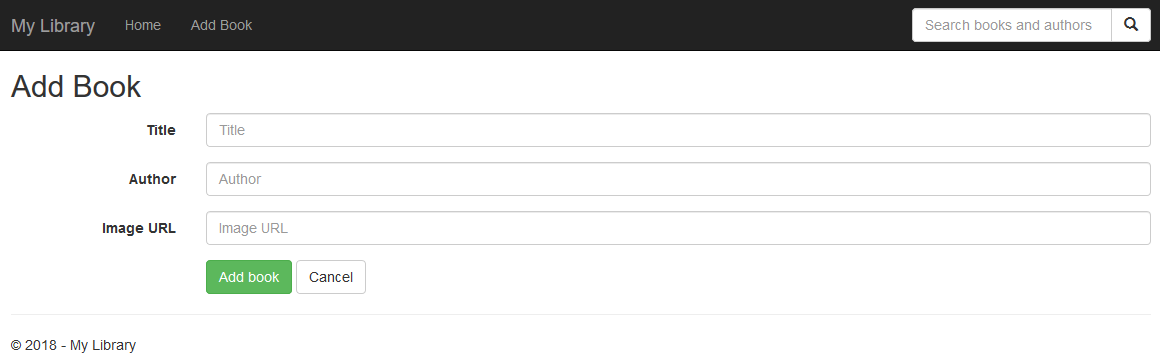
## Add Book

Create a page **/Books/Add** which displays a form. If you’re using images for the book cover, save them as URLs – it’s much easier. We’ll talk about file uploads later.

Create a **POST** handler which saves the book in the database. If you don’t want to, don’t bother with validation. Assume that all data in the form will be valid.

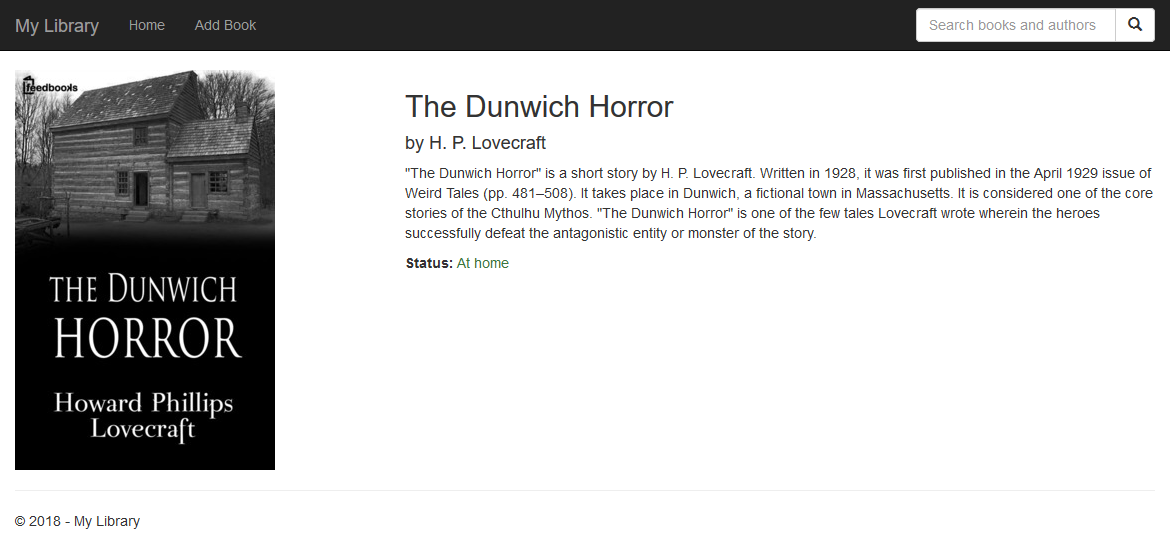
The author field is a freeform text. If the author doesn’t yet exist in the database, create it. Associate the author to the current book.

When you successfully add a book, redirect to its details page.



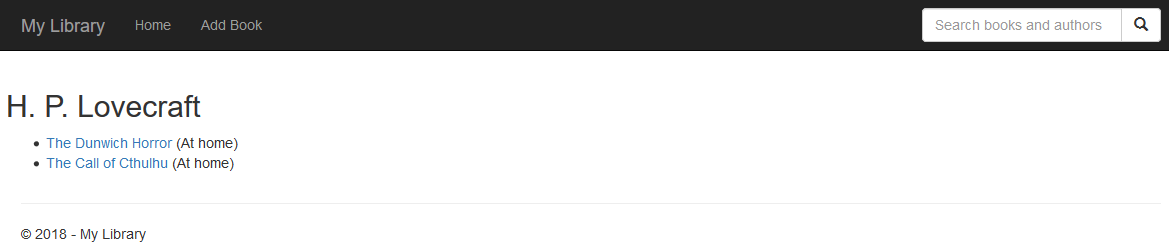
## Book Details

Create a page **/Books/Details/{id}** which displays more information about the book.



## Display All Books by Author

Create a page **/Authors/Details/{id}** which lists all books by the specified author. Every book title should be a link to its details page.



## Search

Create a page **/Search?searchTerm=…** When you type something, return all books whose titles and / or author names contain the search query. The search must be case-insensitive.

