

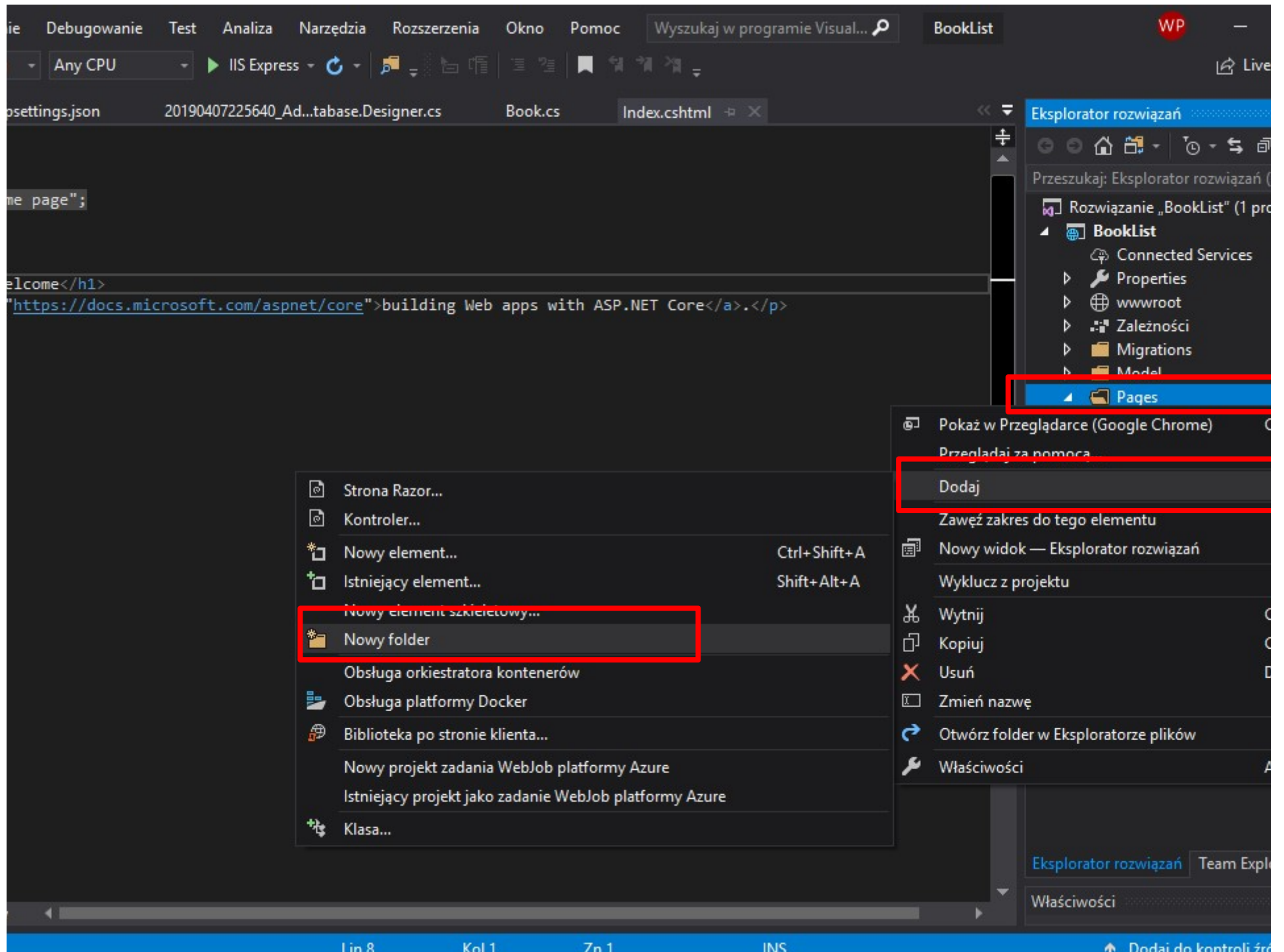
**Creating index page**

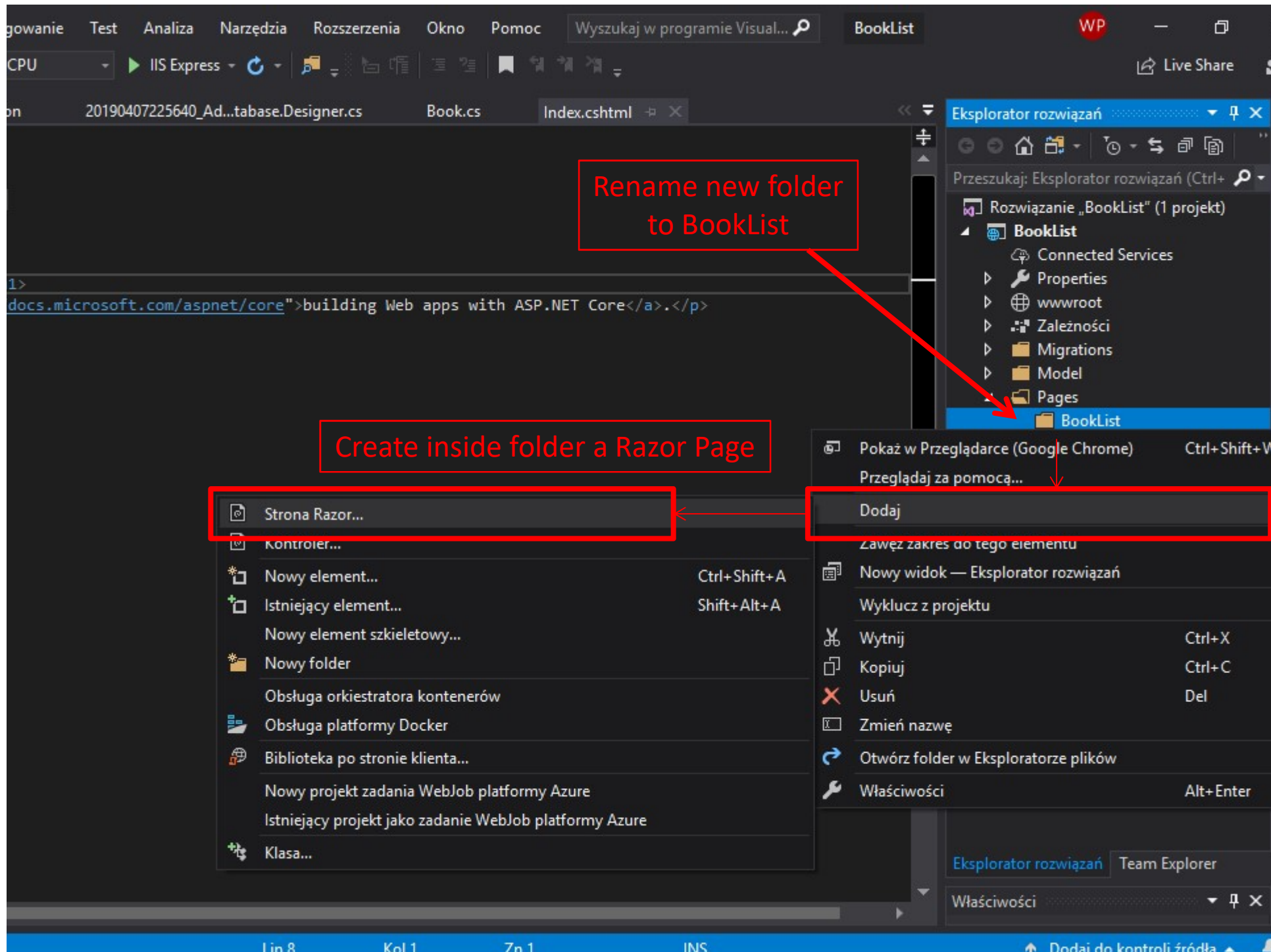
BookList Home Privacy

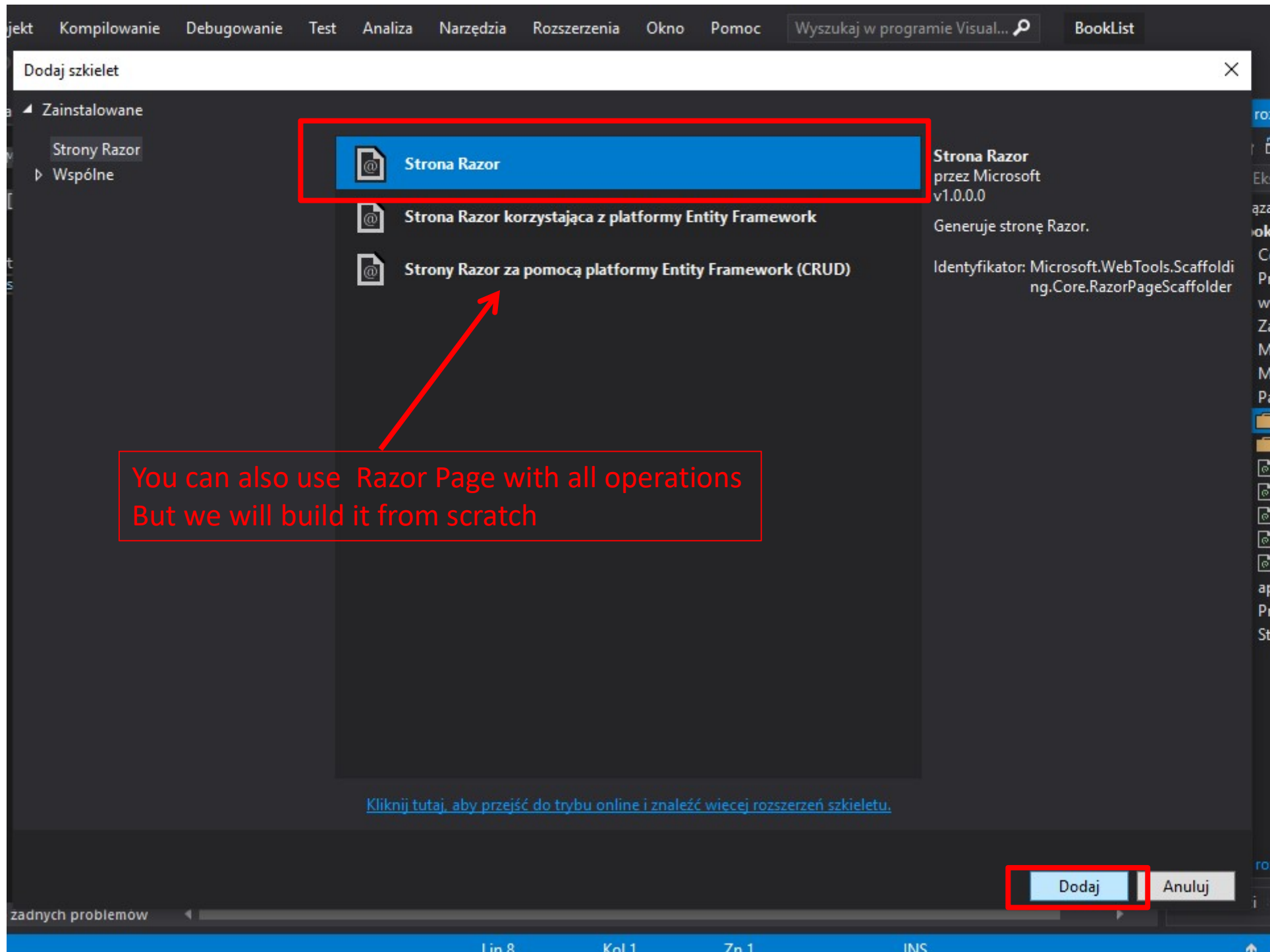
# Welcome

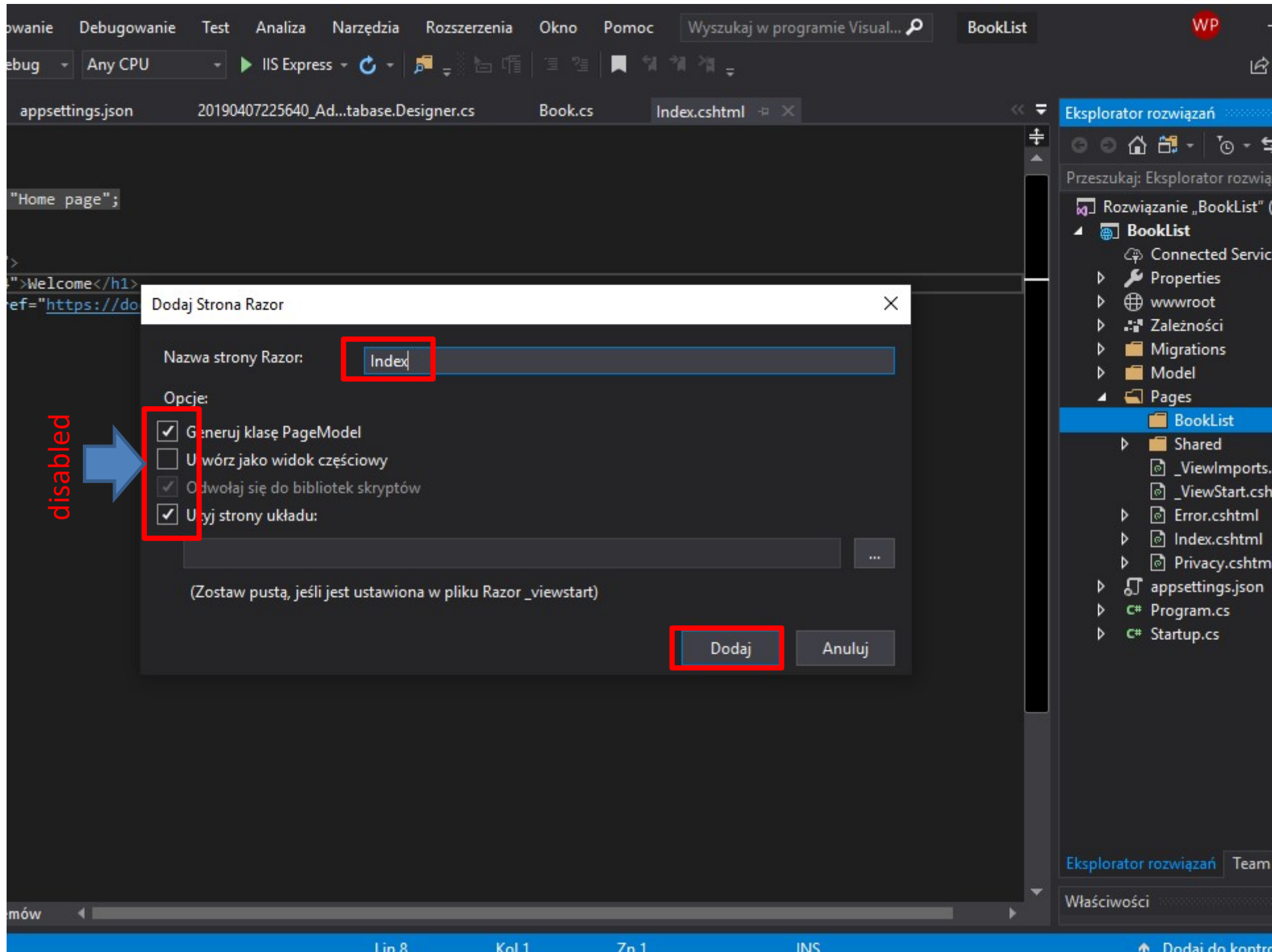
Learn about [building Web apps with ASP.NET Core](#).

Remove „Home” & „Privacy”.  
Insert „Book”









# New index page

```
Index.cshtml | _Layout.cshtml | Index.cshtml.cs | 20190407225640_AddBookToDatabase.cs | Index.cshtml
```

```
1 @page
2 @model BookList.IndexModel
3 @{
4     ViewData["Title"] = "Index";
5 }
6
7 <h1>Book List</h1>
8
9
```

Path to the model

We have to change the header of the page to „Book List”

# Model for index page

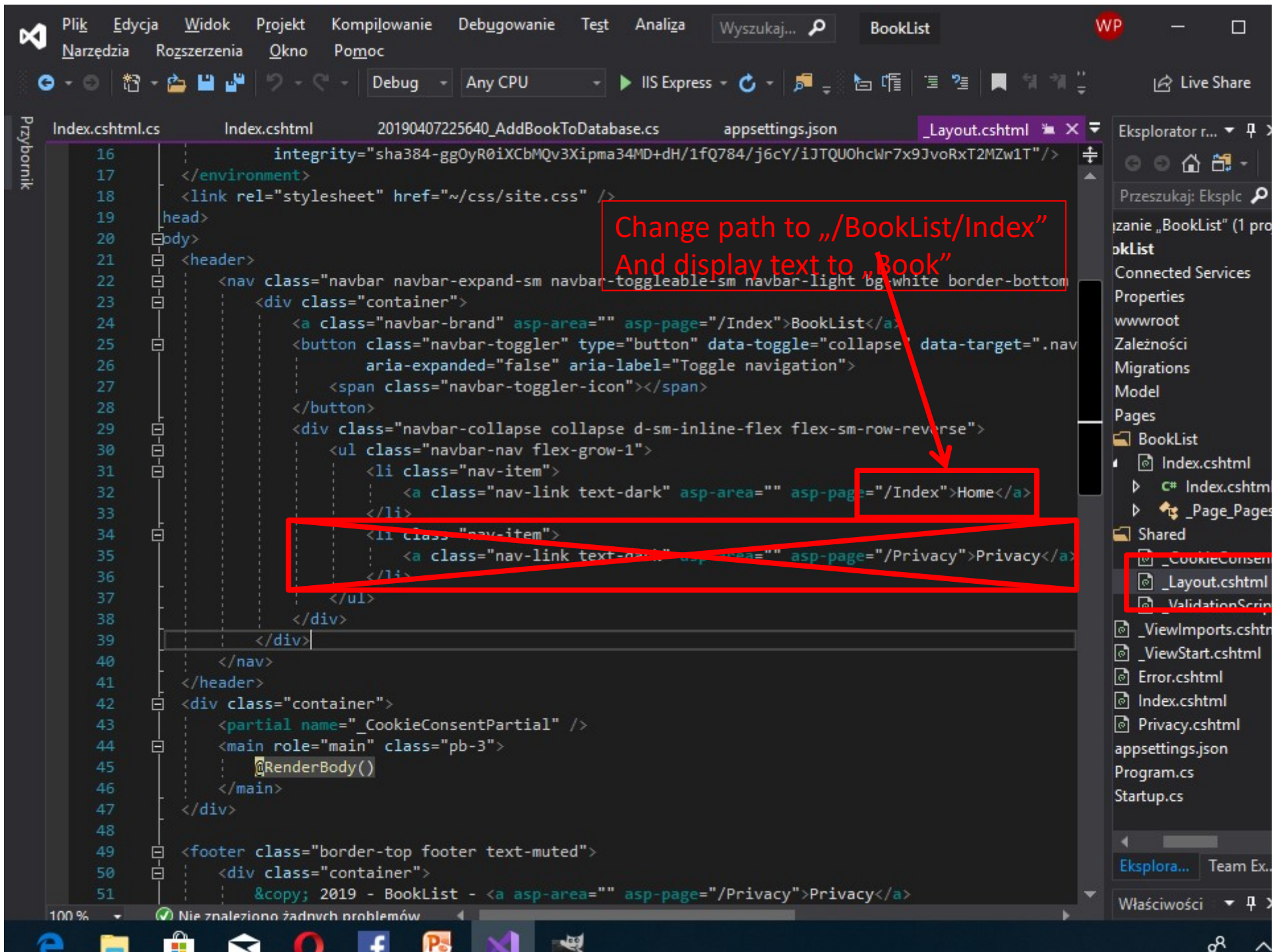
```
Index.cshtml.cs | Index.cshtml | 20190407225640_AddBookToDatabase.cs | appsettings.json
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using Microsoft.AspNetCore.Mvc;
6 using Microsoft.AspNetCore.Mvc.RazorPages;
7
8 namespace BookList.Pages.BookList
9 {
10     public class IndexModel : PageModel
11     {
12         public void OnGet()
13         {
14         }
15     }
16 }
17
```

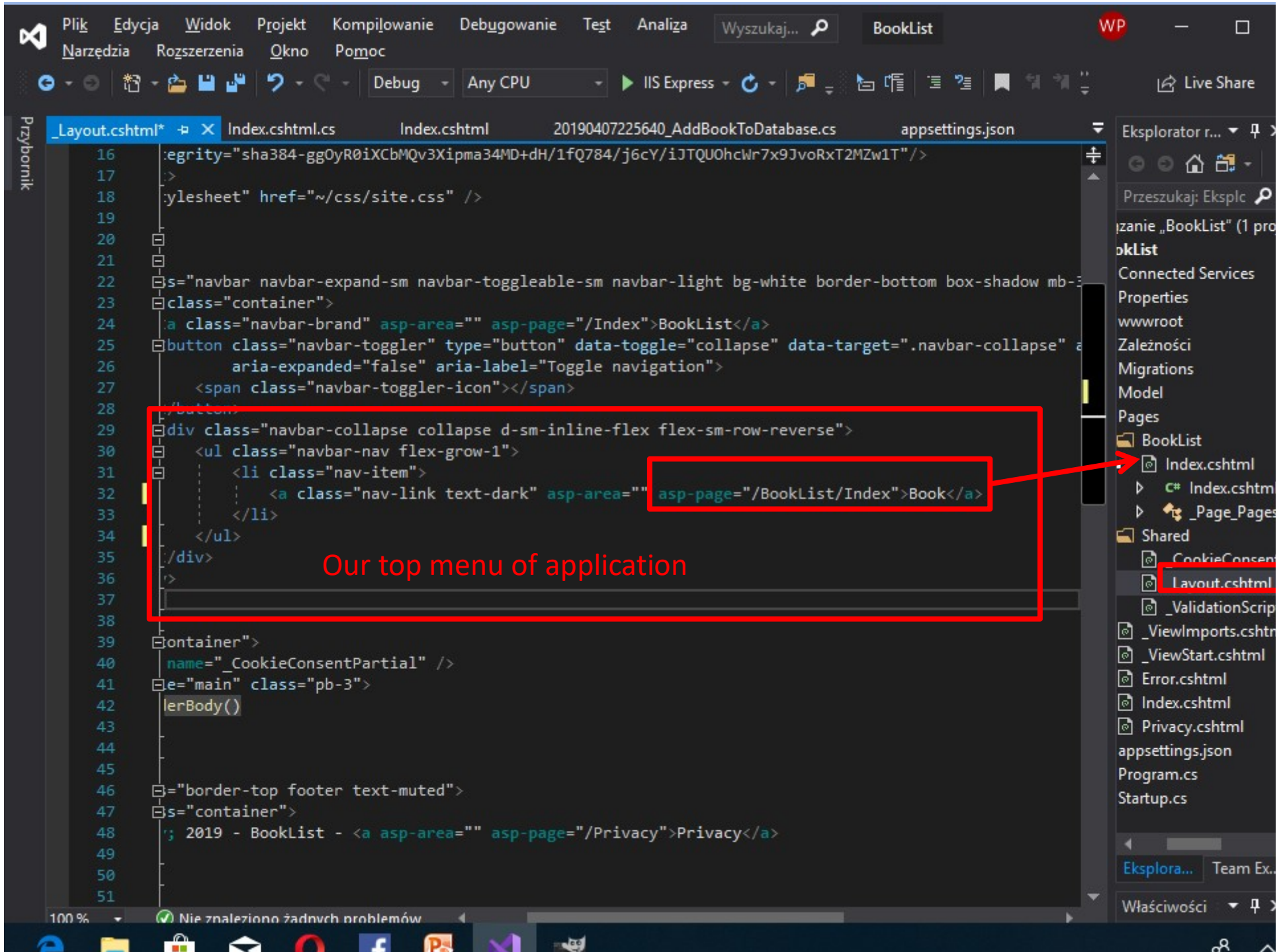
BookList

BookList.Pages.BookList.IndexModel

OnGet()







BookList

Book

click



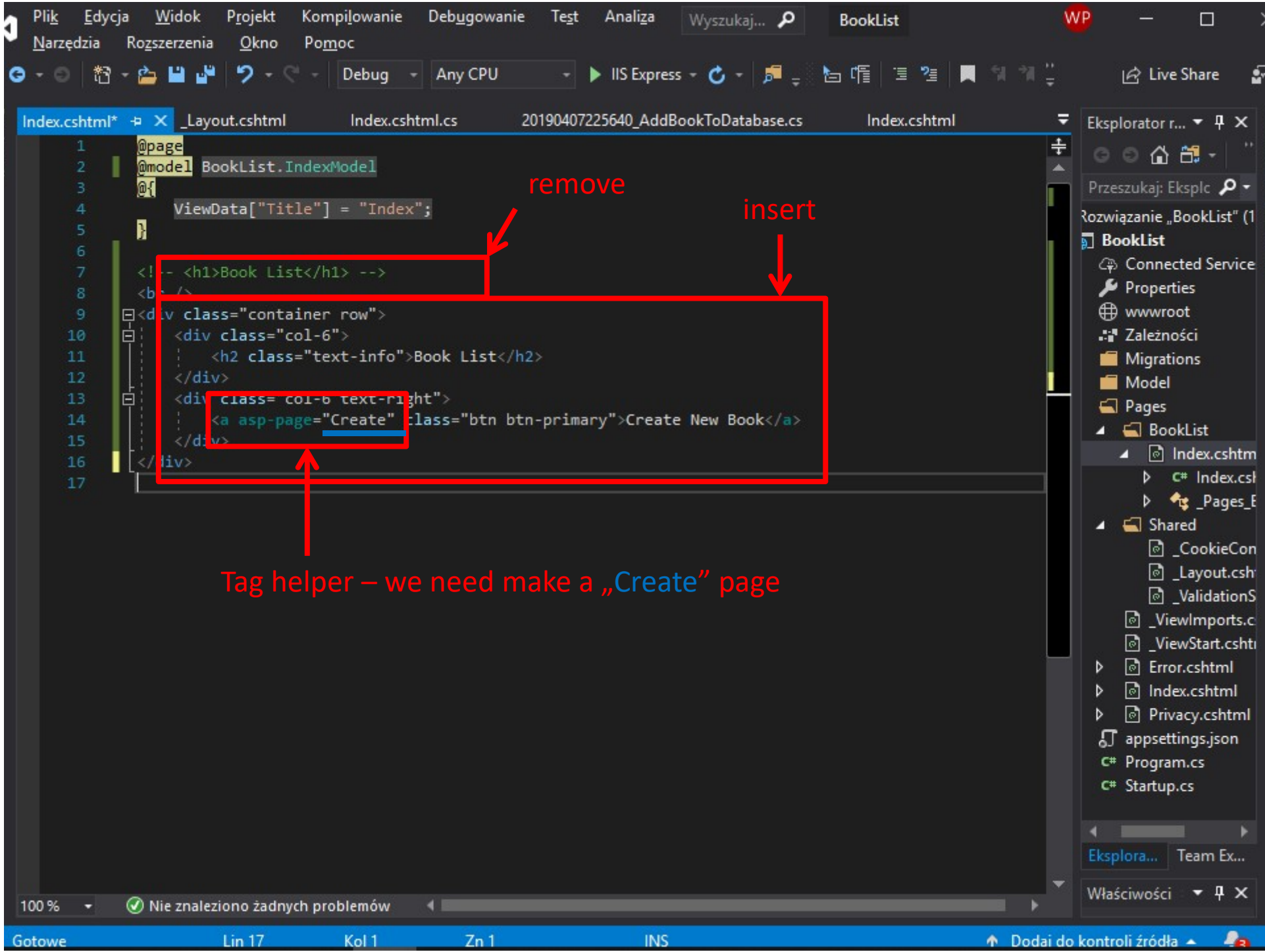
# Welcome

Learn about [building Web apps with ASP.NET Core](#).



# Book List

**„Create New Book” button**



Visual Studio interface showing a Razor view file (`Index.cshtml`) and a context menu for adding a new page.

The code in the editor is as follows:

```
1 @page
2 @model BookList.IndexModel
3 @f
4 ViewData["Title"] = "Index";
5
6
7 <!-- <h1>Book List</h1> -->
8 <br />
9 <div class="container row">
10 <div class="col-6">
11 <h2 class="text-info">Book List</h2>
12 </div>
13 <div class="col-6 text-right">
14 <a asp-page="Create" class="btn btn-primary">Create New Book</a>
15 </div>
16 </div>
17
```

The context menu is open, showing options for adding a new page. The "Dodaj" (Add) option is highlighted in red. The "Strona Razor..." (Razor Page...) option is also highlighted in red.

The Explorer pane on the right shows the project structure, with the "Pages" folder expanded and the "BookList" folder selected. The "Dodaj" option is also highlighted in red.

The status bar at the bottom shows "Nie znaleziono żadnych prob" (No errors found).

Dodaj szkielet



▲ Zainstalowane

Strony Razor

▸ Wspólne



**Strona Razor**



**Strona Razor korzystająca z platformy Entity Framework**



**Strony Razor za pomocą platformy Entity Framework (CRUD)**

**Strona Razor**

przez Microsoft  
v1.0.0.0

Generuje stronę Razor.

Identyfikator: Microsoft.WebTools.Scaffolding.Core.RazorPageScaffolder

[Kliknij tutaj, aby przejść do trybu online i znaleźć więcej rozszerzeń szkieletu.](#)

Dodaj

Anuluj

Plik Edycja Widok Projekt Kompilowanie Debugowanie Test Analiza Wyszukaj... BookList WP

Narzędzia Rozszerzenia Okno Pomoc

Debug Any CPU IIS Express Live Share

Index.cshtml \_Layout.cshtml Index.cshtml.cs 20190407225640\_AddBookToDatabase.cs Index.cshtml

```
1 @page
2 @model BookList.IndexModel
3 @{
4     ViewData["Title"] = "Index";
5 }
6
7 <!-- <h1>Book List </h1>
8 <br />
9 <div class="
10 <div cla
11 <h2
12 </div>
13 <div cla
14 <a as
15 </div>
16 </div>
17
```

**Dodaj Strona Razor**

Nazwa strony Razor:

Opcje:

- Generuj klasę PageModel
- Utwórz jako widok częściowy
- Odwołaj się do bibliotek skryptów
- Użyj strony układu:

...

(Zostaw pustą, jeśli jest ustawiona w pliku Razor \_viewstart)

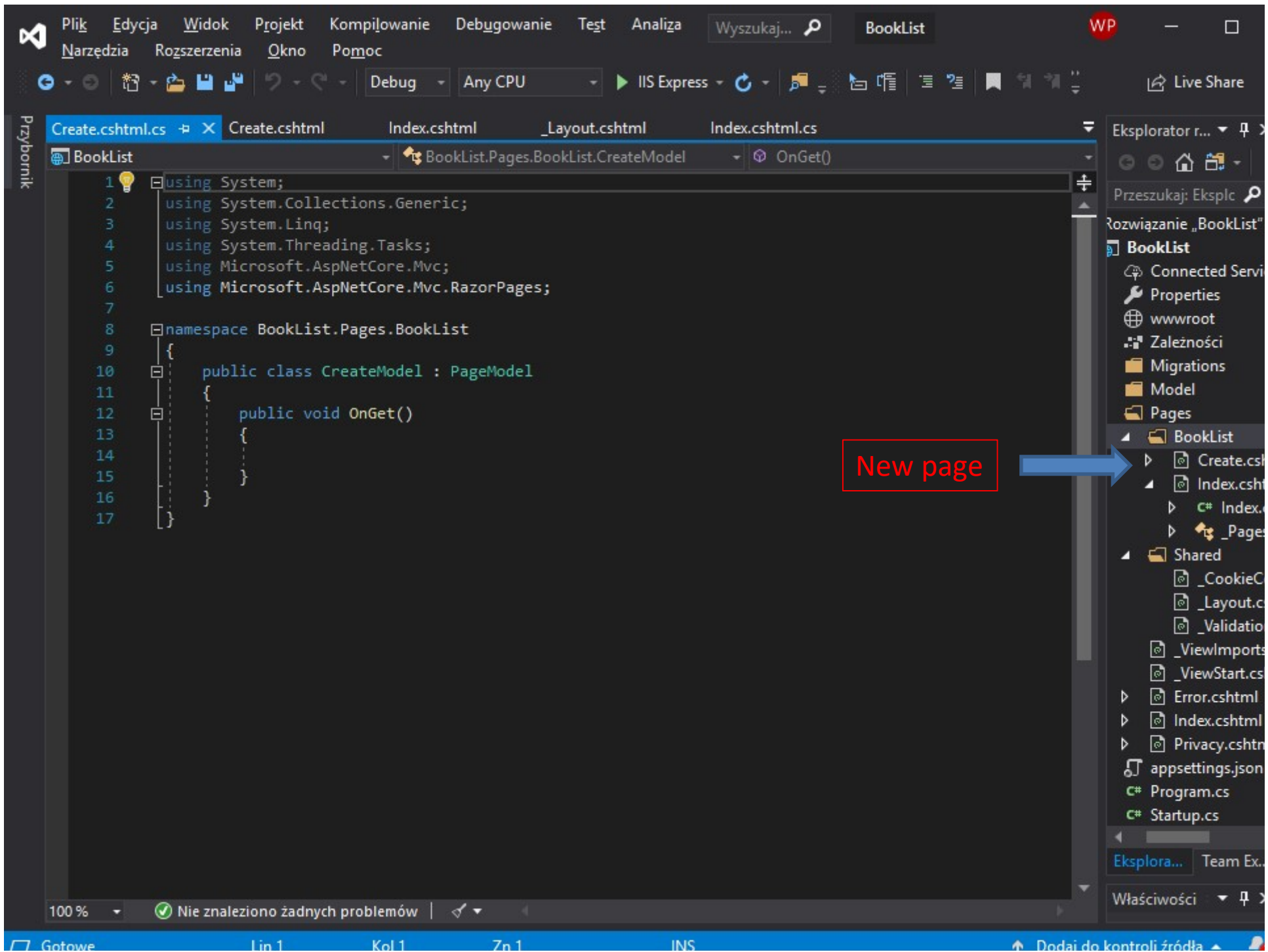
Eksplo... Team Ex...

Właściwości

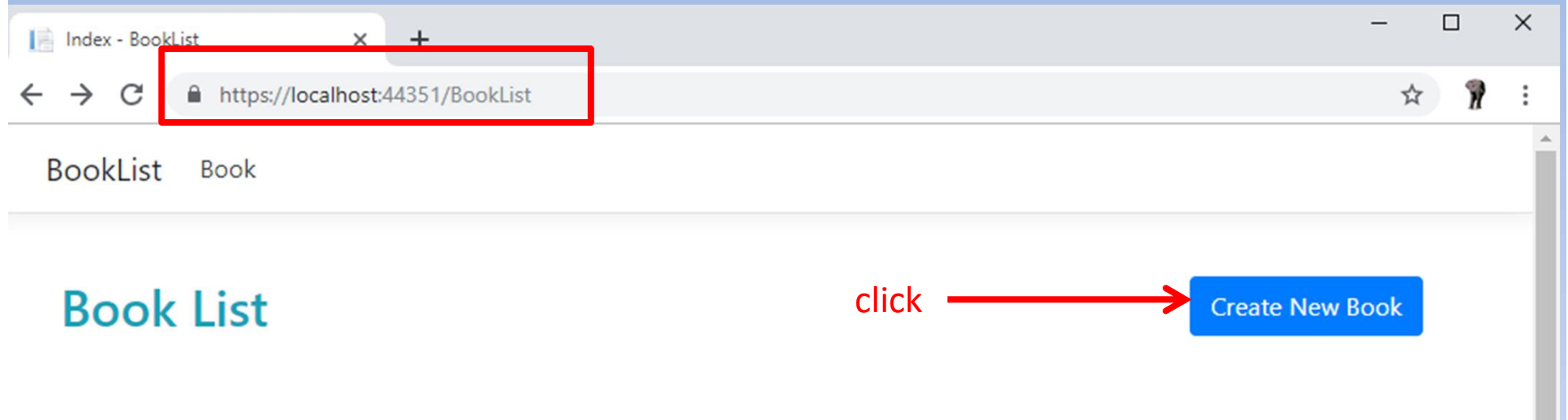
100% Nie znaleziono żadnych problemów

Ten element nie obsłua... Lin 17 Kol 1 Zn 1 INS Dodaj do kontroli źródła

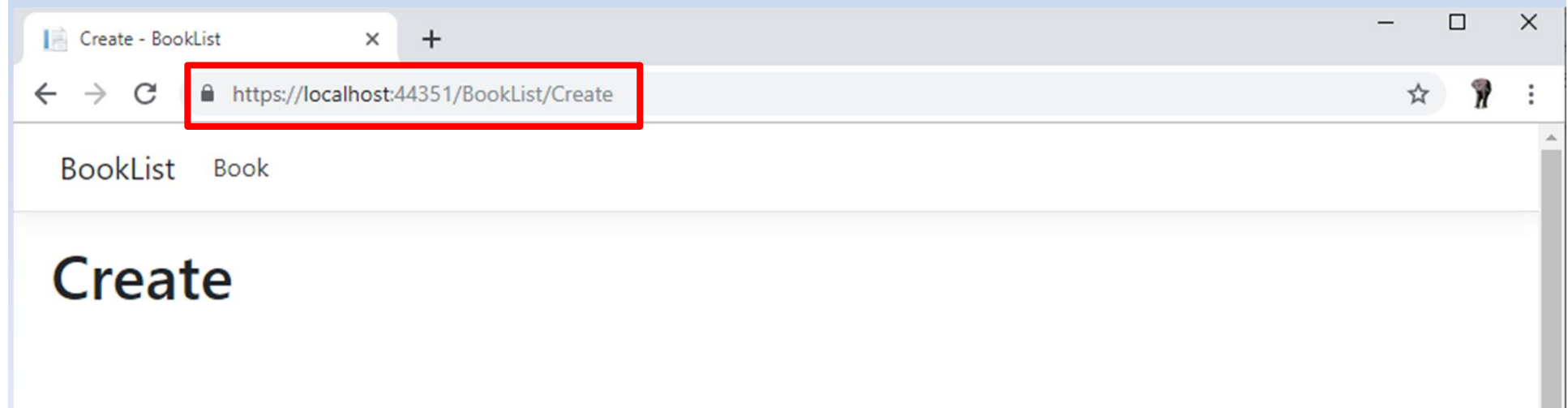




# „Index” page



# „Create” page



**Bootstrap classes**

```
1 @page  
2 @model BookList.CreateModel  
3 @f  
4 ViewData["Title"] = "Create";  
5  
6  
7 <!-- <h1>Create</h1> -->  
8  
9 <br />  
10 <h2 class="text-info">Create New Book</h2>  
11 <br />  
12  
13 <div class="border">  
14 <form method="post">  
15 <div asp-validation-summary="ModelOnly" class="text-danger"></div>  
16 <div class="container" style="padding:30px;">  
17 <div class="form-group">  
18 <div class="row">  
19 <div class="col-sm-2">  
20  
21 </div>  
22 <div class="col-sm-4">  
23  
24 </div>  
25 </div>  
26 </div>  
27 </form>  
28 </div>  
29  
30
```

**Form validation – Book.Name is required**

**Page in a bootstrap is divided into 12 col's. It will occupy half a page**

**Eksplorator rozwiązań**  
Przeszukaj: Eksplorator rozwiązań (Ctrl+;)  
Rozwiązanie „BookList” (1 projekt)  
BookList  
Connected Services  
Properties  
wwwroot  
Zależności  
Migrations  
Model  
Pages  
BookList  
Create.cshtml  
Create.cshtml.cs  
Pages\_BookList\_Create  
Index.cshtml  
Index.cshtml.cs  
Shared  
\_CookieConsentPartial.cshtml  
\_Layout.cshtml  
\_ValidationScriptsPartial.cshtml  
\_ViewImports.cshtml  
\_ViewStart.cshtml  
Error.cshtml  
Index.cshtml  
Privacy.cshtml  
appsettings.json  
Program.cs

## Create New Book

Half a page for our row

```
7 namespace BookList.Model
8 {
9     public class Book
10    {
11        [Key]
12        public int Id { get; set; }
13        [Required]
14        [Display(Name = "Book Name")]
15        public string Name { get; set; }
16    }
17 }
18
19 using System.Threading.Tasks;
20 using BookList.Model;
21 using Microsoft.AspNetCore.Mvc;
22 using Microsoft.AspNetCore.Mvc.RazorPages;
23
24 namespace BookList.Pages.BookList
25 {
26     public class CreateModel : PageModel
27     {
28         [BindProperty]
29         public Book Book { get; set; }
30
31         public void OnGet()
32         {
33         }
34     }
35 }
36
37 <div asp-validation-summary="ModelOnly" class="text-danger" ></div>
38 <div class="container" style="padding:30px;" >
39     <div class="form-group" >
40         <div class="row" >
41             <div class="col-sm-2" >
42                 <label asp-for="Book.Name" ></label>
43             </div>
44             <div class="col-sm-4" >
45                 <input asp-for="Book.Name" class="form-control" />
46             </div>
47         </div>
48     </div>
49 </div>
```

## Create New Book

Book Name

Menu: Plik, Edycja, Widok, Projekt, Kompilowanie, Debugowanie, Test, Analiza, Wyszukaj...  
Narzędzia, Rozszerzenia, Okno, Pomoc

Debug, Any CPU, IIS Express, Live Share

Open tabs: Create.cshtml.cs, \_Layout.cshtml, Index.cshtml.cs, Book.cs, Create.cshtml\*

```
13 <div class="border">
14   <form method="post">
15     <div asp-validation-summary="ModelOnly" class="text-danger"></div>
16     <div class="container" style="padding:30px;">
17       <div class="form-group">
18         <div class="row">
19           <div class="col-sm-2">
20             <label asp-for="Book.Name"></label>
21           </div>
22           <div class="col-sm-4">
23             <input asp-for="Book.Name" class="form-control" />
24           </div>
25         </div>
26       </div>
27       <div class="form-group">
28         <div class="row">
29           <div class="col-sm-2">
30             <label asp-for="Book.Author"></label>
31           </div>
32           <div class="col-sm-4">
33             <input asp-for="Book.Author" class="form-control" />
34           </div>
35         </div>
36       </div>
37       <div class="form-group">
38         <div class="row">
39           <div class="col-sm-2">
40             <label asp-for="Book.ISBN"></label>
41           </div>
42           <div class="col-sm-4">
43             <input asp-for="Book.ISBN" class="form-control" />
44           </div>
45         </div>
46       </div>
47     </div>
48   </form>
49 </div>
```

Annotations: Red box around lines 17-26, red underlines for Book.Name, Book.Author, and Book.ISBN, red arrows pointing from the box to the underlines.

100% Nie znaleziono żadnych problemów

Gotowe Lin 43 Kol 33 Zn 33 INS

Eksplorator rozwiązań

Przeszukaj: Eksplorator rozwiązań (Ctrl+;)

- Rozwiązanie „BookList” (1 projekt)
  - BookList
    - Connected Services
    - Properties
    - wwwroot
    - Zależności
    - Migrations
    - Model
      - ApplicationDbContext.cs
      - Book.cs
    - Pages
      - BookList
        - Create.cshtml
          - Create.cshtml.cs
          - \_Pages\_BookList\_Create
        - Index.cshtml
          - Index.cshtml.cs
      - Shared
        - \_CookieConsentPartial.cshtml
        - \_Layout.cshtml
        - \_ValidationScriptsPartial.cshtml
        - \_ViewImports.cshtml
        - \_ViewStart.cshtml
      - Error.cshtml
      - Index.cshtml
      - Privacy.cshtml

Eksplorator rozwiązań Team Explorer

Właściwości

## Create New Book

Book Name

Author

ISBN

Visual Studio interface showing the code for `Create.cshtml` in the `BookList` project. The code defines a form with two input fields and two buttons.

```
29 <div class="col-sm-2">
30   <label asp-for="Book.Author"></label>
31 </div>
32 <div class="col-sm-4">
33   <input asp-for="Book.Author" class="form-control" />
34 </div>
35 </div>
36 </div>
37 <div class="form-group">
38   <div class="row">
39     <div class="col-sm-2">
40       <label asp-for="Book.ISBN"></label>
41     </div>
42     <div class="col-sm-4">
43       <input asp-for="Book.ISBN" class="form-control" />
44     </div>
45   </div>
46 </div>
47 <div class="form-group">
48   <div class="row">
49     <div class="col-sm-2 offset-sm-2">
50       <input type="submit" value="Create" class="btn btn-primary form-control" />
51     </div>
52     <div class="col-sm-2">
53       <a asp-page="Index" class="btn btn-success form-control" >Back to list</a>
54     </div>
55   </div>
56 </div>
57 </div>
58 >
59 </div>
60 </div>
```

A red box highlights the two buttons in the code, and a red arrow points to them with the text "Two buttons".

The right sidebar shows the `Eksplorator rozwiązań` (Solution Explorer) for the `BookList` project, with `Create.cshtml` selected.

At the bottom, the status bar shows "100%" zoom, "Nie znaleziono żadnych problemów" (No problems found), and the current line is 52, column 41.



## Create New Book

Book Name

Author

ISBN

Click „Create” button – nothing happens

**Create post handler**

The image shows a screenshot of the Visual Studio IDE with the following components:

- Menu Bar:** Plik, Edycja, Widok, Projekt, Kompilowanie, Debugowanie, Test, Analiza, Wyszukaj..., BookList, WP.
- Toolbar:** Narzędzia, Rozszerzenia, Okno, Pomoc, Debug, Any CPU, IIS Express, Live Share.
- Editor:** Shows the code for `Create.cshtml.cs` in the `BookList.Pages.BookList.CreateModel` namespace. The code includes:

```
6 using Microsoft.AspNetCore.Mvc;
7 using Microsoft.AspNetCore.Mvc.RazorPages;
8
9 namespace BookList.Pages.BookList
10 {
11     public class CreateModel : PageModel
12     {
13         private readonly ApplicationDbContext _db;
14         public CreateModel(ApplicationDbContext db)
15         {
16             _db = db;
17         }
18
19         [BindProperty]
20         public Book Book { get; set; }
21
22         public void OnGet()
23         {
24         }
25
26         public async Task<IActionResult> OnPost()
27         {
28             if (!ModelState.IsValid)
29             {
30                 return Page();
31             }
32             _db.Book.Add(Book);
33             await _db.SaveChangesAsync();
34             return RedirectToPage("Index");
35         }
36     }
37 }
38
39
```
- Annotations:** Three red boxes highlight specific code sections with blue arrows pointing to them:
  - Box 1:** Encloses the `private readonly ApplicationDbContext _db;` field and the `public CreateModel(ApplicationDbContext db) { _db = db; }` constructor. A red box next to it contains the text: "DB Context in CreateModel class".
  - Box 2:** Encloses the `public void OnGet()` method. A red box next to it contains the text: "Here we can make changes on page load".
  - Box 3:** Encloses the `public async Task<IActionResult> OnPost()` method. A red box next to it contains the text: "If the form is not correct – return the same page".
- Box 4:** A red box next to the `return RedirectToPage("Index");` line in the `OnPost` method contains the text: "Else – add Book and go to Index page".

The right-hand side of the image shows the **Eksplorator rozwiązań** (Solution Explorer) with the project structure for `BookList`. The `Pages` folder is expanded, and the `Create.cshtml` file is highlighted.

At the bottom of the IDE, the status bar shows "100%", "Nie znaleziono żadnych problemów", and "Lin 36 Kol 44 Zn 44 INS".

# Create New Book

Book Name

Author

ISBN

click

```
public async Task<IActionResult> OnPost()  
{  
    if (!ModelState.IsValid) ≤ 1 ms upłynęło  
        return View();  
    // ...  
}
```

Count	3
ErrorCount	1
HasReachedMaxErrors	false
IsValid	false
Keys	{Microsoft.AspNetCore.Mvc.ModelBinding.ModelStateDictionary.KeyE...}
MaxAllowedErrors	200
Root	SubKey={}, Key="", ValidationState=Unvalidated
ValidationState	Invalid
Values	{Microsoft.AspNetCore.Mvc.ModelBinding.ModelStateDictionary.Valu...}
Non-Public members	
Results View	Expanding the Results View will enumerate the IEnumerable
[0]	SubKey={ISBN}, Key="Book.ISBN", ValidationState=Valid
[1]	SubKey={Name}, Key="Book.Name", ValidationState=Invalid
[2]	SubKey={Author}, Key="Book.Author", ValidationState=Valid

Podsumowanie Zdarzenia Użycie pamięci Uży...


Zdarzenia

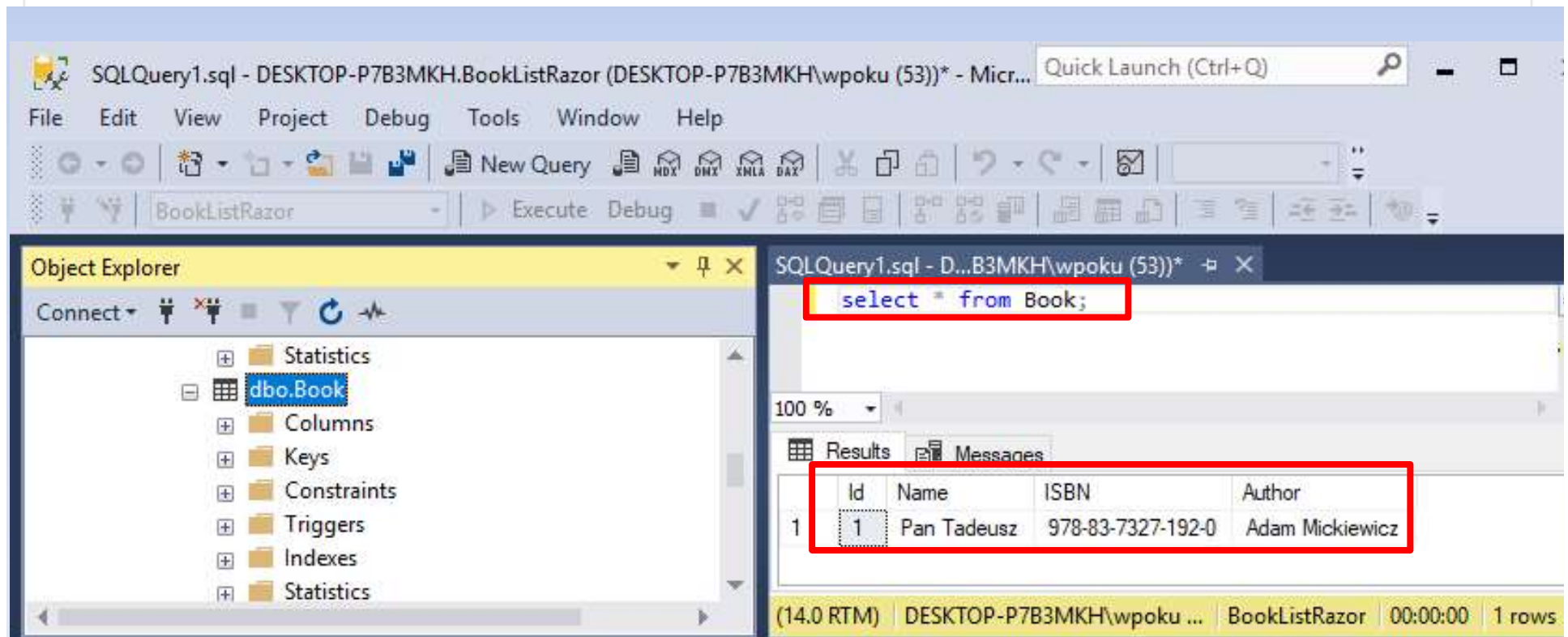
# Create New Book

Book Name

Author

ISBN

click 



The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left shows the database structure, with the `dbo.Book` table selected. The SQL Query window on the right contains the query `select * from Book;`. The Results pane shows the following data:

	Id	Name	ISBN	Author
1	1	Pan Tadeusz	978-83-7327-192-0	Adam Mickiewicz

The status bar at the bottom indicates that the query returned 1 row.

# Validation

```
11 <br />
12
13 <div class="border">
14   <form method= post >
15     <div asp-validation-summary="ModelOnly" class="text-danger"></div>
16     <div class="container" style="padding: 20px;">
17       <div class="form-group">
18         <div class="row">
19           <div class="col-sm-2">
20             <label asp-for="Book.Name"></label>
21           </div>
22           <div class="col-sm-4">
23             <input asp-for="Book.Name" class="form-control" />
24           </div>
25           <span asp-validation-for="Book.Name" class="text-danger"></span>
26         </div>
27       </div>
28     </form>
29   </div>
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61 @section Scripts{
62   @await Html.RenderPartialAsync("_ValidationScriptsPartial");
63 }
64
65
```

Book Name

Author

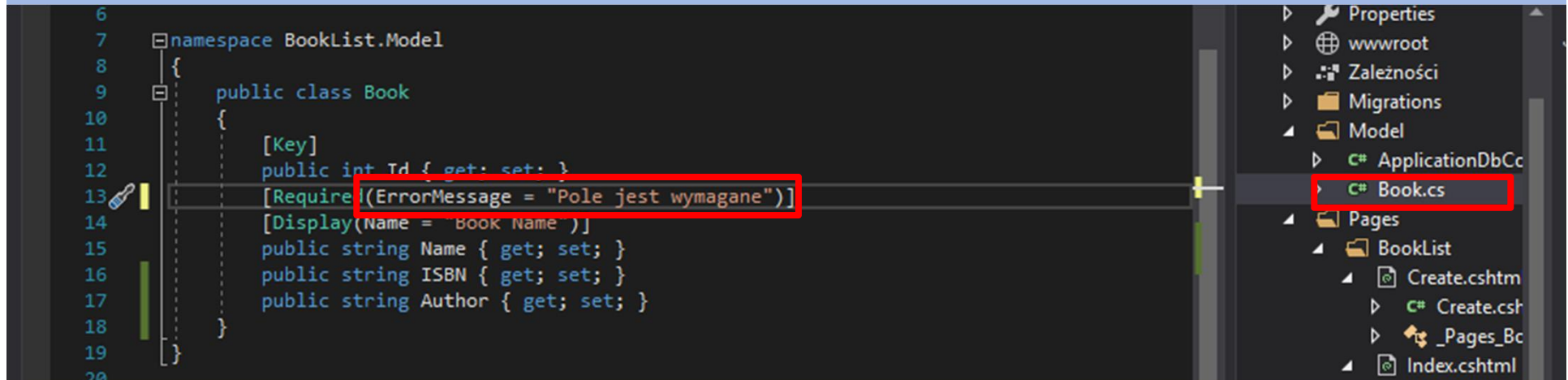
ISBN

The Book Name field is required.

How to change the text?

# Error message change

```
6
7 namespace BookList.Model
8 {
9     public class Book
10    {
11        [Key]
12        public int Id { get; set; }
13        [Required(ErrorMessage = "Pole jest wymagane")]
14        [Display(Name = "BOOK Name")]
15        public string Name { get; set; }
16        public string ISBN { get; set; }
17        public string Author { get; set; }
18    }
19 }
20
```

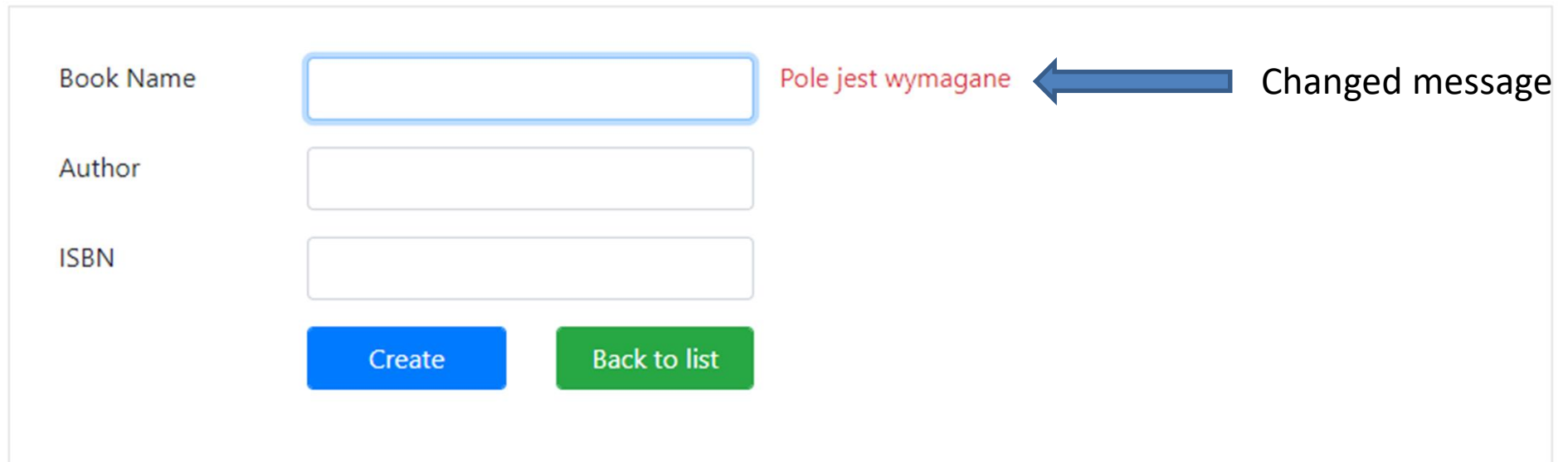


## Create New Book

Book Name  Pole jest wymagane ← Changed message

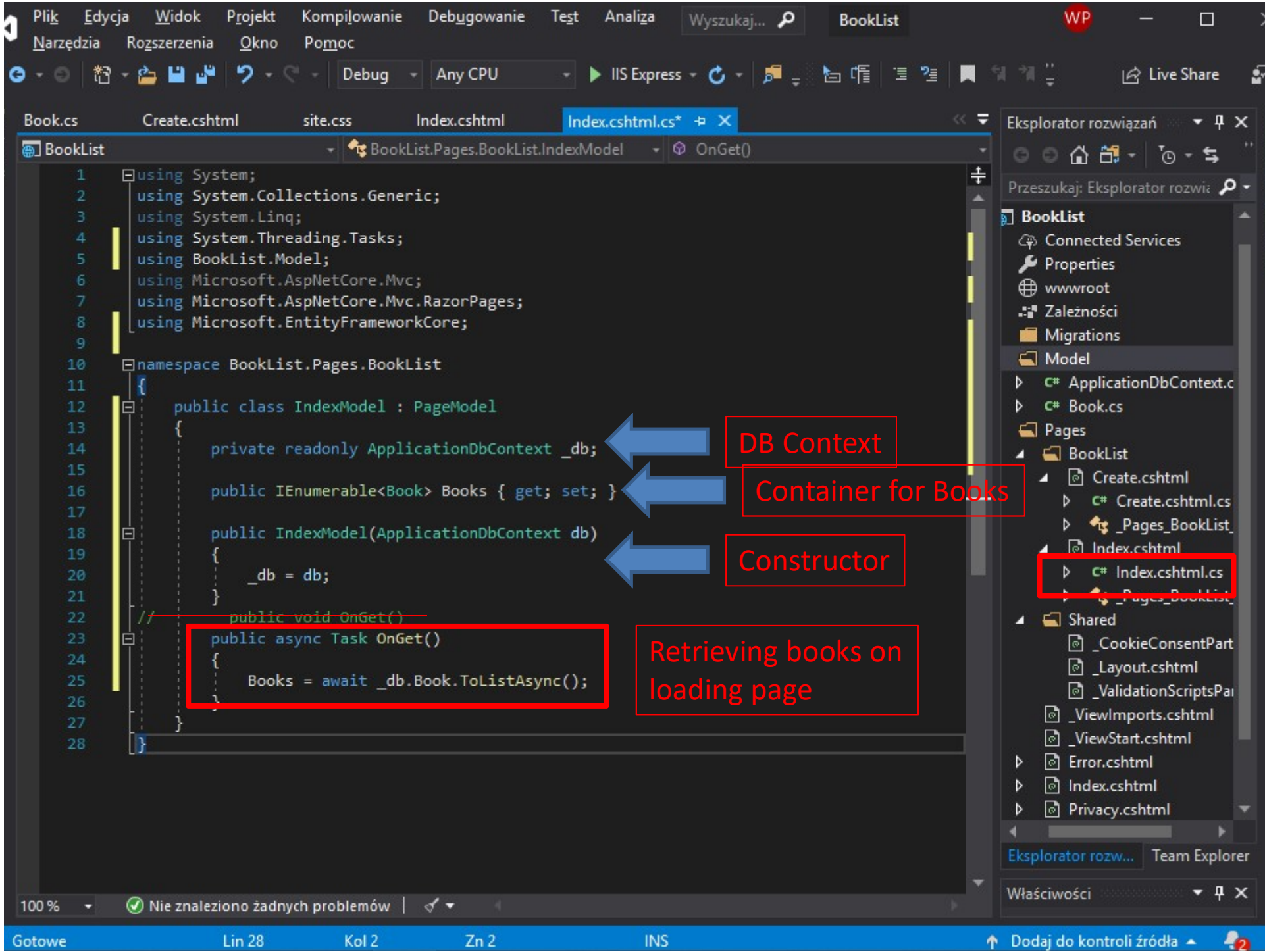
Author

ISBN



**Retrieving „Books” from DB**





DB Context

Container for Books

Constructor

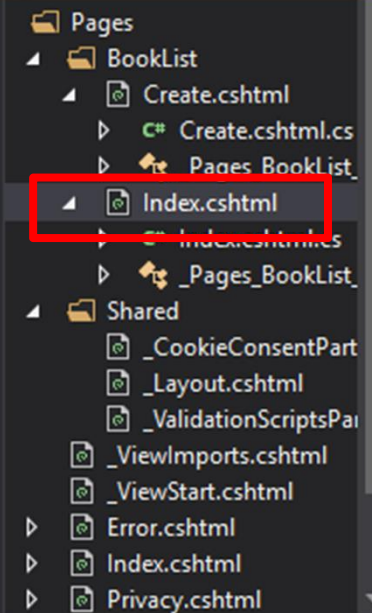
Retrieving books on loading page

# Table header

```
17
18 @if (Model.Books.Count() > 0)
19 {
20     <br /><br />
21     <table class="table table-striped border">
22         <tr class="table-secondary">
23             <th>
24                 @Html.DisplayNameFor(m => m.Books.FirstOrDefault().Name)
25             </th>
26             <th>
27                 @Html.DisplayNameFor(m => m.Books.FirstOrDefault().Author)
28             </th>
29             <th>
30                 @Html.DisplayNameFor(m => m.Books.FirstOrDefault().ISBN)
31             </th>
32         </tr>
33     </table>
34 }
35 else
36 {
37     <p>
38         No Books Available.
39     </p>

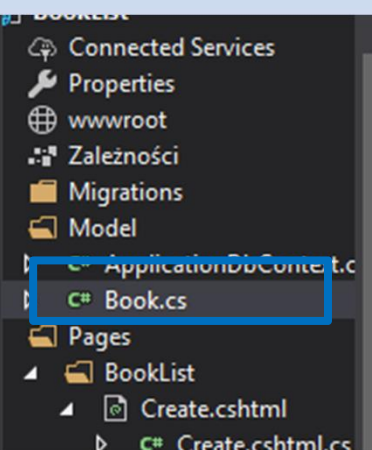
```

Displays header  
„Book Name”  
from DB



```
7 namespace BookList.Model
8 {
9     public class Book
10     {
11         [Key]
12         public int Id { get; set; }
13         [Required(ErrorMessage = "Pole jest wymagane")]
14         [Display(Name = "Book Name")]
15         public string Name { get; set; }
16         public string ISBN { get; set; }
17         public string Author { get; set; }
18     }
19 }

```



# Book List

Create New Book

**Book Name**

**Author**

**ISBN**

```
27     @Html.DisplayNameFor(m => m.Books.FirstOrDefault().Author)
28     </th>
29     <th>
30         @Html.DisplayNameFor(m => m.Books.FirstOrDefault().ISBN)
31     </th>
32     </tr>
33     @foreach (var item in Model.Books)
34     {
35         <tr>
36             <td>
37                 @Html.DisplayFor(m => item.Name)
38             </td>
39             <td>
40                 @Html.DisplayFor(m => item.Author)
41             </td>
42             <td>
43                 @Html.DisplayFor(m => item.ISBN)
44             </td>
45         </tr>
46     }
47 </table>
48 }
```

Displays value From DB

BookList Book

## Book List

Create New Book

Book Name	Author	ISBN
Pan Tadeusz	Adam Mickiewicz	978-83-7327-192-0

**Edit & Delete buttons**

The image shows a Visual Studio IDE window for a project named "BookList". The main editor displays the file "Index.cshtml" with the following code:

```
28     </th>
29     <th>
30         @Html.DisplayNameFor(m => m.Books.FirstOrDefault().ISBN)
31     </th>
32     <th></th>
33 >
34     @foreach (var item in Model.Books)
35     {
36         <tr>
37             <td>
38                 @Html.DisplayFor(m => item.Name)
39             </td>
40             <td>
41                 @Html.DisplayFor(m => item.Author)
42             </td>
43             <td>
44                 @Html.DisplayFor(m => item.ISBN)
45             </td>
46             <td>
47                 <button class="btn btn-danger btn-sm">Delete</button>
48                 <a asp-route-id="@item.Id" asp-page="Edit" class="btn btn-success btn-sm">Edit</a>
49             </td>
50         </tr>
51     }
52 >
53     <div>
54         @if (Model.Count == 0)
55         {
56             <p>No books Available.</p>
57         }
58     }
59 </div>
```

A red rectangular box highlights the code on lines 46-49, specifically the "Edit" link: `<a asp-route-id="@item.Id" asp-page="Edit" class="btn btn-success btn-sm">Edit</a>`. A blue callout box with a white border and text "We need to pass „id” into the edit page" has a blue arrow pointing to the `@item.Id` expression within the `asp-route-id` attribute.

The right-hand side of the IDE shows the "Eksplorator rozwiązań" (Solution Explorer) with the project structure for "BookList". The "Index.cshtml" file is selected and highlighted in blue.

At the bottom of the IDE, the status bar shows "100%", "Nie znaleziono żadnych problemów" (No problems found), and "Lin 32 Kol 17 Zn 17 INS".

# Book List

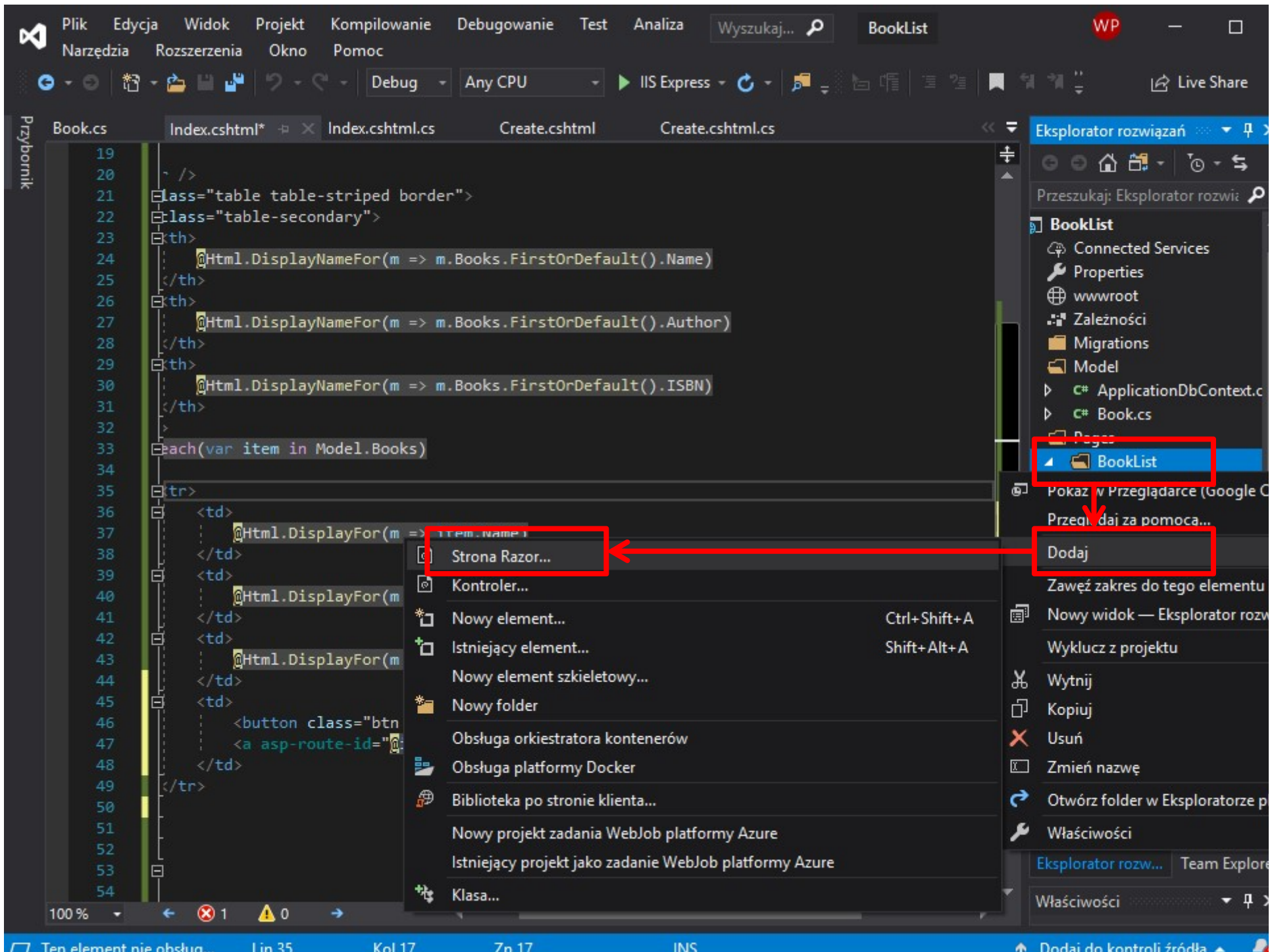
Create New Book

Book Name	Author	ISBN	
Pan Tadeusz	Adam Mickiewicz	978-83-7327-192-0	<a href="#">Delete</a> <a href="#">Edit</a>

New buttons

**Edit page**





Plik Edycja Widok Projekt Kompilowanie Debugowanie Test Analiza Wyszukaj... BookList WP

Narzędzia Rozszerzenia Okno Pomoc

Debug Any CPU IIS Express Live Share

Book.cs Index.cshtml\* Index.cshtml.cs Create.cshtml Create.cshtml.cs

```
19 </>
20 </>
21 <table class="table table-striped border">
22 <table class="table-secondary">
23 <thead>
24 <th>@Html.DisplayNameFor(m => m.Books.FirstOrDefault().Name)
25 </th>
26 <th>
27 <th>@Html.Dis
28 </th>
29 <th>
30 <th>@Html.Dis
31 </th>
32 </thead>
33 <tbody>
34 <tr>
35 <tr>
36 <td>
37 <td>@Html
38 </td>
39 <td>
40 <td>@Html
41 </td>
42 <td>
43 <td>@Html
44 </td>
45 <td>
46 <button class="btn btn-danger btn-sm">Delete</button>
47 <a asp-route-id="@item.id" asp-page="Edit" class="btn btn-success btn-sm">Edit</a>
48 </td>
49 </tr>
50
51
52
53
54
```

**Dodaj Strona Razor**

Nazwa strony Razor: Edit

Opcje:

- Generuj klasę PageModel
- Utwórz jako widok częściowy
- Odwołaj się do bibliotek skryptów
- Użyj strony układu:

(Zostaw pustą, jeśli jest ustawiona w pliku Razor\_viewstart)

Dodaj Anuluj

Eksplorator rozwiązań

Przeszukaj: Eksplorator rozwiązań

BookList

- Connected Services
- Properties
- wwwroot
- Zależności
- Migrations
- Model
- C# ApplicationDbContext.cs
- C# Book.cs
- Pages
  - BookList
    - Create.cshtml
      - C# Create.cshtml.cs
      - \_Pages\_BookList\_
    - Index.cshtml
      - C# Index.cshtml.cs
      - \_Pages\_BookList\_
  - Shared
    - \_CookieConsentPart
    - \_Layout.cshtml
    - \_ValidationScriptsPa
    - \_ViewImports.cshtml
    - \_ViewStart.cshtml
  - Error.cshtml
  - Index.cshtml
  - Privacy.cshtml

Właściwości Team Explorer

100% 1 0

Visual Studio interface showing the code for `EditModel.cshtml`. The code includes the following using statements:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using BookList.Model;
6 using Microsoft.AspNetCore.Mvc;
7 using Microsoft.AspNetCore.Mvc.RazorPages;
```

The `namespace BookList.Pages.BookList` contains the `public class EditModel : PageModel` class. The class has a private `ApplicationDbContext _db` and a constructor `public EditModel(ApplicationDbContext db)` that sets `_db = db`.

The `public Book Book { get; set; }` property is highlighted with a red box, and a red arrow points to it from a red box labeled "Property to fill".

The `OnGet(int id)` method is defined as follows:

```
23 public void OnGet(int id)
24 {
25     Book = _db.Book.Find(id);
26 }
```

The Solution Explorer on the right shows the project structure for `BookList`, including the `Pages` folder and the `BookList` folder.

Visual Studio interface showing the code editor for `Edit.cshtml` and the Solution Explorer on the right.

The code editor displays the following HTML and Razor code for the `Edit.cshtml` page:

```
1 @page
2 @model BookList.Pages.BookList.EditModel
3 @f
4 ViewData["Title"] = "Edit";
5
6
7 <!-- <h1>Edit</h1> -->
8
9 <br />
10 <h2 class="text-info">Create New Book</h2>
11 <br />
12
13 <div class="border">
14   <form method="post">
15     <div asp-validation-summary="ModelOnly" class="text-danger"></div>
16     <div class="container" style="padding:30px;">
17       <div class="form-group">
18         <div class="row">
19           <div class="col-sm-2">
20             <label asp-for="Book.Name"></label>
21           </div>
22           <div class="col-sm-4">
23             <input asp-for="Book.Name" class="form-control" />
24           </div>
25           <span asp-validation-for="Book.Name" class="text-danger"></span>
26         </div>
27       </div>
28       <div class="form-group">
29         <div class="row">
30           <div class="col-sm-2">
31             <label asp-for="Book.Author"></label>
32           </div>
33           <div class="col-sm-4">
34             <input asp-for="Book.Author" class="form-control" />
35           </div>
36         </div>
37       </div>
38     </div>
39   </form>
40 </div>
```

The Solution Explorer on the right shows the project structure for `BookList`. The `Create.cshtml` file is highlighted with a red box, and a red arrow points to it from a text box that says "Copy all from „Create” page".

At the bottom of the window, the status bar shows "Lin 28 Kol 2 Zn 2 INS".

Visual Studio interface showing the code for an ASP.NET Core MVC application. The code is in `Edit.cshtml` and includes annotations explaining key parts:

- The name of page**: Points to the `<!-- <h1>Edit</h1> -->` comment.
- BindProperty from page model**: Points to the `asp-for="Book.Id"` attribute in the `<input type="hidden" asp-for="Book.Id" />` tag.
- The same content**: Points to the `<input type="submit" value="Update" />` button.
- „Update” instead of „Create”**: Points to the `value="Update"` attribute of the submit button.

```
1 @page
2 @model BookList.EditModel
3
4 ViewData["Title"] = "Edit";
5
6
7 <!-- <h1>Edit</h1> -->
8
9 <br />
10 <h2 class="text-info">Edit Book</h2>
11 <br />
12
13 <div class="border">
14 <form method="post">
15 <div class="text-danger"></div>
16 <input type="hidden" asp-for="Book.Id" />
17 <div class="container" style="padding: 50px;">
18 <div class="form-group">...</div>
19 <div class="form-group">...</div>
20 <div class="form-group">...</div>
21 <div class="form-group">
22 <div class="row">
23 <div class="col-sm-2 offset-sm-2">
24 <input type="submit" value="Update" class="btn btn-primary form-control" />
25 </div>
26 <div class="col-sm-2">
27 <a asp-page="Index" class="btn btn-success form-control">Back to 1
28 </div>
29 </div>
30 </div>
31 </div>
32 </form>
33 </div>
34
35 @section Scripts{
36 @await Html.RenderPartialAsync("_ValidationScriptsPartial");
37 }
```

100% Nie znaleziono żadnych problemów

Gotowe Lin 61 Kol 7 Zn 7 INS Dodaj do kontroli źródła

Browser tabs: Edit - BookList

Address bar: <https://localhost:44351/BookList/Edit?id=1>

BookList Book

## Edit Book

Book Name: Pan Tadeusz

Author: Adam Mickiewicz

ISBN: 978-83-7327-192-0

Update Back to list

Pre-filled fields

We have to write a post-handler for our „Update” button

© 2019 - BookList - Privacy

The image shows a Visual Studio IDE window with the following components:

- Menu Bar:** Plik, Edycja, Widok, Projekt, Kompilowanie, Debugowanie, Test, Analiza, Wyszukaj..., BookList, WP.
- Toolbar:** Narzędzia, Rozszerzenia, Okno, Pomoc, Debug, Any CPU, IIS Express, Live Share.
- Editor:** Edit.cshtml, Startup.cs, ApplicationDbContext.cs, Edit.cshtml.cs, Program.cs. The code in Edit.cshtml.cs is as follows:

```
11 public class EditModel : PageModel
12 {
13     private ApplicationDbContext _db;
14
15     public EditModel(ApplicationDbContext db)
16     {
17         _db = db;
18     }
19
20     [BindProperty]
21     public Book Book { get; set; }
22
23     public async Task OnGet(int id)
24     {
25         Book = await _db.Book.FindAsync(id);
26     }
27
28     public async Task<IActionResult> OnPost()
29     {
30         if (!ModelState.IsValid)
31         {
32             return RedirectToPage();
33         }
34
35         var BookFromDb = await _db.Book.FindAsync(Book.Id);
36
37         BookFromDb.Name = Book.Name;
38         BookFromDb.Author = Book.Author;
39         BookFromDb.ISBN = Book.ISBN;
40
41         await _db.SaveChangesAsync();
42
43         return RedirectToPage("Index");
44     }
45 }
```
- File Explorer (Eksplorator rozwiązań):** Shows the project structure with 'Edit.cshtml.cs' highlighted in a red box.
- Annotations:** A red box highlights the OnPost method. Three blue arrows point from red text boxes to specific lines of code:
  - 'Redirect to self' points to line 32: `return RedirectToPage();`
  - 'Save to DB' points to lines 37-39: `BookFromDb.Name = Book.Name;`, `BookFromDb.Author = Book.Author;`, and `BookFromDb.ISBN = Book.ISBN;`
  - 'Redirect to Index' points to line 43: `return RedirectToPage("Index");`
- Status Bar:** 100%, Nie znaleziono żadnych problemów, Lin 44, Kol 10, Zn 10, INS, Dodaj do kontroli źródła.

## Edit Book

Book Name

Author

ISBN

Update

Back to list



BookList Book

# Book List

[Create New Book](#)

Book Name	Author	ISBN	
<u>Pan Tadeusz - wyd. poprawione</u>	Adam Mickiewicz	978-83-7327-192-0	<a href="#">Delete</a> <a href="#">Edit</a>

**Delete book**

Visual Studio interface showing the code for `Index.cshtml` in a project named `BookList`. The code is in C# and Razor syntax, defining a form for creating a new book and a table for displaying existing books.

```
12 </div>
13 <div class="col-6 text-right">
14 <a asp-page="Create" class="btn btn-primary">Create New Book</a>
15 </div>
16 </div>
17
18 <form method="post">
19     if (Model.Books.Count() > 0)
20     {
21         <br /><br />
22         <table class="table table-striped border">
23             <tr class="table-secondary">...</tr>
24             <thead>
25                 <tr>
26                     <th>Title</th>
27                     <th>Author</th>
28                     <th>Year</th>
29                     <th>Actions</th>
30                 </tr>
31             </thead>
32             <tbody>
33                 <foreach (var item in Model.Books)>
34                 <tr>
35                     <td>...</td>
36                     <td>...</td>
37                     <td>...</td>
38                     <td>
39                         <button asp-page-handler="Delete" asp-route-id="@item.Id" class="btn btn-danger">Delete</button>
40                         <a asp-route-id="@item.Id" asp-page="Edit" class="btn btn-success">Edit</a>
41                     </td>
42                 </tr>
43             </tbody>
44         </table>
45     }
46     else
47     {
48         <p>No Books Available.</p>
49     }
50 </form>
```

The code includes several annotations:

- A red box highlights the `<form method="post">` tag.
- A red box highlights the `asp-page-handler="Delete"` attribute on the button.
- A red box highlights the `asp-route-id="@item.Id"` attribute on the button.
- A red box highlights the `asp-route-id="@item.Id"` attribute on the link.
- A red box highlights the `</form>` closing tag.
- A blue arrow labeled "Route Id" points from the `asp-route-id="@item.Id"` attribute on the button to the `Index.cshtml` file in the Solution Explorer.

The Solution Explorer on the right shows the project structure, with `Index.cshtml` highlighted in a red box. The status bar at the bottom indicates "100%" zoom and "Nie znaleziono żadnych problemów" (No problems found).

Visual Studio interface showing the code for the `IndexModel` class in the `BookList` project. The code is displayed in the main editor window, and the file explorer on the right shows the project structure.

```
10 namespace BookList.Pages.BookList
11 {
12     public class IndexModel : PageModel
13     {
14         private readonly ApplicationDbContext _db;
15
16         public IEnumerable<Book> Books { get; set; }
17
18         public IndexModel(ApplicationDbContext db)
19         {
20             _db = db;
21         }
22
23         public async Task OnGet()
24         {
25             Books = await _db.Book.ToListAsync();
26         }
27
28         public async Task<IActionResult> OnPostDelete(int id)
29         {
30             var book = await _db.Book.FindAsync(id);
31             if (book == null)
32             {
33                 return NotFound();
34             }
35
36             _db.Book.Remove(book);
37
38             await _db.SaveChangesAsync();
39
40             return RedirectToPage("Index");
41         }
42     }
43 }
```

The code is annotated with red boxes and arrows:

- Route Id**: A red box pointing to the `id` parameter in the `OnPostDelete(int id)` method signature.
- Remove from DB**: A red box pointing to the `_db.Book.Remove(book);` line.
- Make changes**: A red box pointing to the `await _db.SaveChangesAsync();` line.
- Refresh page**: A red box pointing to the `return RedirectToPage("Index");` line.

The file explorer on the right shows the project structure, with the `Index.cshtml.cs` file highlighted in red.

# Book List

Create New Book

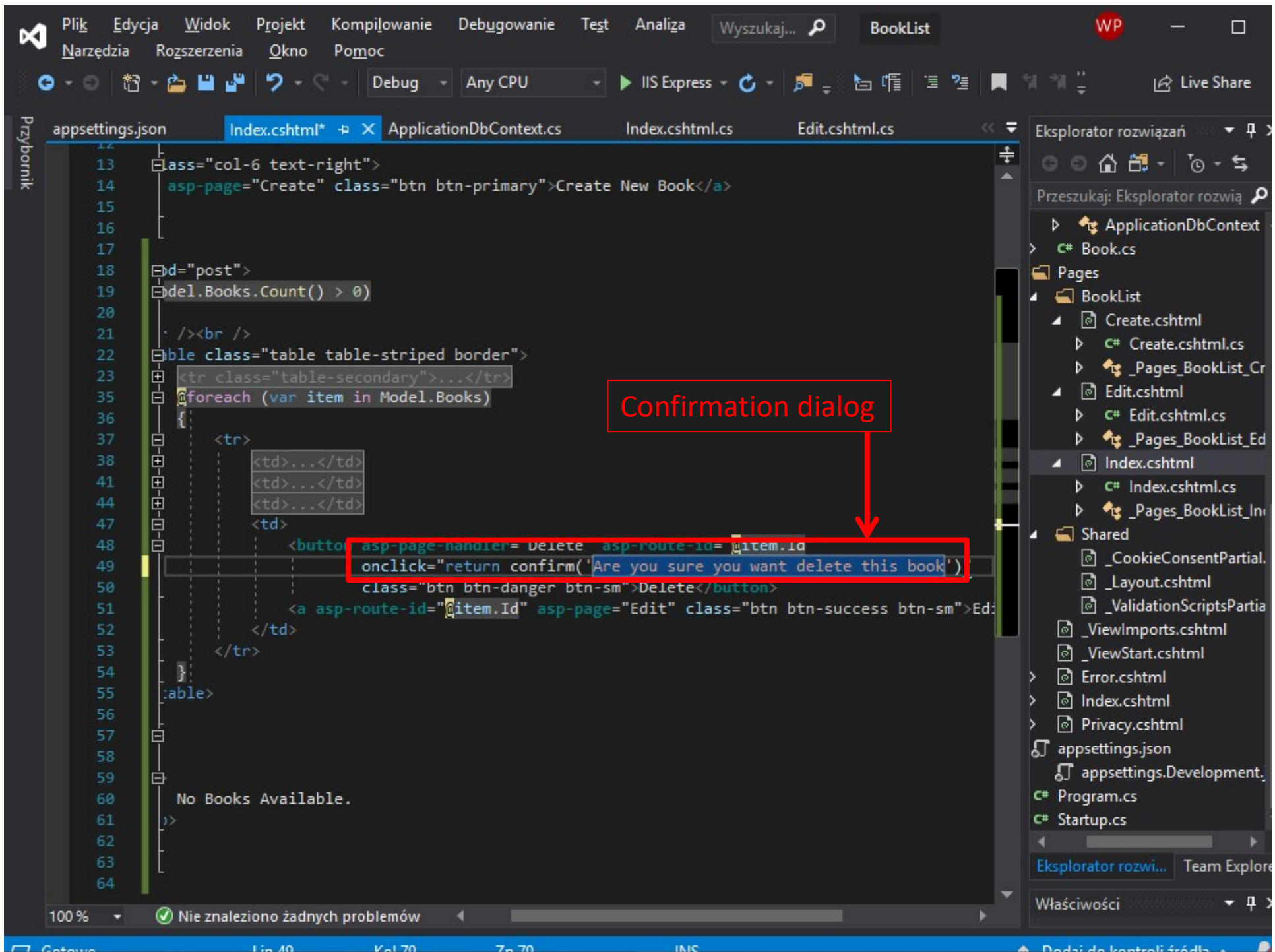
Book Name	Author	ISBN	
Pan Tadeusz - wyd. poprawione	Adam Mickiewicz	978-83-7327-192-0	Delete Edit
Book 1	Author 1	1234	Delete Edit
Book 2	Author 2	5678	click Delete Edit



# Book List

Create New Book

Book Name	Author	ISBN	
Pan Tadeusz - wyd. poprawione	Adam Mickiewicz	978-83-7327-192-0	Delete Edit
Book 1	Author 1	1234	Delete Edit



# Book List

Komunikat ze strony localhost:44351  
Are you sure you want delete this book

OK

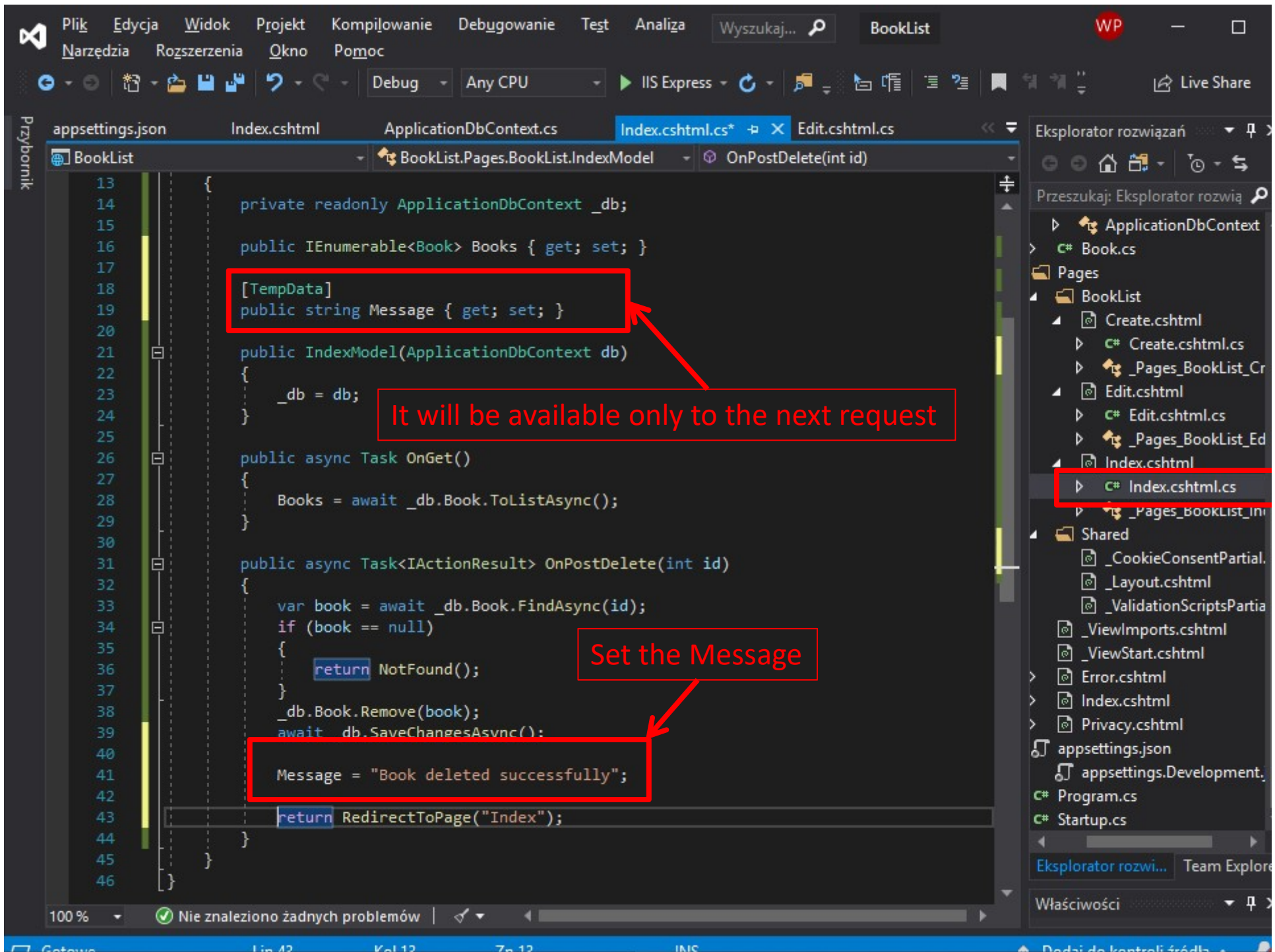
Anuluj

[Create New Book](#)

Book Name	Author	ISBN	
Pan Tadeusz - wyd. poprawione	Adam Mickiewicz	978-83-7327-192-0	<a href="#">Delete</a> <a href="#">Edit</a>
Book 1	Author 1	1234	<a href="#">Delete</a> <a href="#">Edit</a>

**Message after delete**





```
17
18 @if(Model.Message != null)
19 {
20     <div class="alert alert-info alert-dismissible" role="alert">
21         <button type="button" class="close" data-dismiss="alert" aria-label="close">
22             <span aria-hidden="true">&times;</span>
23         </button>
24         @Model.Message
25     </div>
26 }
27
28 <form method="post">
29     @if (Model.Books.Count() > 0)
30     {
31         <br /><br />
32         <table class="table table-striped border">
33             <tr class="table-secondary">...
```

## Book List

Create New Book

Book deleted successfully

Message after delete

Book Name	Author	ISBN	
Pan Tadeusz - wyd. poprawione	Adam Mickiewicz	978-83-7327-192-0	Delete Edit

**Message after create**

Visual Studio interface showing the code for `Create.cshtml.cs` in a project named `BookList`. The code defines a `CreateModel` class with a constructor, a `Book` property, a `Message` property, and an `OnPost` method. The `OnPost` method checks for model validity, saves the book, and redirects to the index page with a success message.

```
14
15 public CreateModel(ApplicationDbContext db)
16 {
17     _db = db;
18 }
19
20 [BindProperty]
21 public Book Book { get; set; }
22
23 [TempData]
24 public string Message { get; set; }
25
26 public void OnGet()
27 {
28 }
29
30
31 public async Task<IActionResult> OnPost()
32 {
33     if (!ModelState.IsValid)
34     {
35         return Page();
36     }
37     _db.Book.Add(Book);
38     await db.SaveChangesAsync();
39
40     Message = "Book created successfully";
41
42     return RedirectToPage("Index");
43 }
44
45 }
```

Message after create will active after redirect to index page

## Create New Book

Book Name

Author

ISBN

Create

Back to list

# Book List

[Create New Book](#)

Book created successfully



Book Name	Author	ISBN	
Pan Tadeusz - wyd. poprawione	Adam Mickiewicz	978-83-7327-192-0	<a href="#">Delete</a> <a href="#">Edit</a>
Book 3	Author 3	9012	<a href="#">Delete</a> <a href="#">Edit</a>

**Message after edit**

Visual Studio interface showing the code for `Edit.cshtml.cs` in the `BookList` project. The code is as follows:

```
18 }
19
20 [BindProperty]
21 public Book Book { get; set; }
22
23 [TempData]
24 public string Message { get; set; }
25
26 public async Task OnGet(int id)
27 {
28     Book = await _db.Book.FindAsync(id);
29 }
30
31 public async Task<IActionResult> OnPost()
32 {
33     if (!ModelState.IsValid)
34     {
35         return RedirectToPage();
36     }
37
38     var BookFromDb = await _db.Book.FindAsync(Book.Id);
39     BookFromDb.Name = Book.Name;
40     BookFromDb.Author = Book.Author;
41     BookFromDb.ISBN = Book.ISBN;
42     await db.SaveChangesAsync();
43
44     Message = "Book updated successfully";
45
46     return RedirectToPage("Index");
47 }
48
49
50 }
```

The `[TempData]` attribute and the `Message` property are highlighted with a red box. The `Message = "Book updated successfully";` line is also highlighted with a red box.

The Solution Explorer on the right shows the project structure, with `Edit.cshtml.cs` selected under `Pages > BookList > Edit.cshtml`.

At the bottom, the status bar shows "100%" zoom, "Nie znaleziono żadnych problemów" (No problems found), and "Lin 49 Kol 2 Zn 2 INS".



## Edit Book

Book Name

Author

ISBN

Update

Back to list

# Book List

[Create New Book](#)

Book updated successfully



Book Name	Author	ISBN	
Pan Tadeusz - wyd. poprawione	Adam Mickiewicz	978-83-7327-192-0	<a href="#">Delete</a> <a href="#">Edit</a>
Book 3	Author 3	9012-456-234	<a href="#">Delete</a> <a href="#">Edit</a>