Web Based Technologies – HW#3

In this homework, you will design a website that displays book information and allows user to add books into shopping cart.

You have to design **four** web pages (**Default.aspx**, **Login.aspx**, **BookInfo.aspx**, and **Cart.aspx**) and a class page (**Book.cs**) that holds book information.

Preliminary Work

1. Create a web project for the homework.
2. Download the sample book-cover images from the web site and put these images into a folder named **images** in your web folder.

Book.cs

1. Add a class file named **Book.cs** into your project (Add New Item, Class). This file will contain information about the books. If you encounter a message which states that this file should be in **App_Code** folder, then select **Yes** and let Visual Studio create the **App_Code** folder for you and put **Book.cs** file in this folder. After then, you will be able to use the **Book** class in your application like any other classes of .NET library.
2. Add member variables to the class **Book** which are **BookID** (integer), **Title** (string), **Author** (string), **Publisher** (string), **PageNumber** (int), and **ImageUrl** (string).
3. Define a constructor for the **Book** class which takes all of its member variables as parameter (**Hint**: In Visual Studio, type **ctor** and press **TAB** key. This constructs an empty constructor for the class. After then, change the constructor parameters and implementation).
4. Finally, your Book class should be like this:

```
public class Book
{
    public int BookID;
    public string Title;
    public string Author;
    public string Publisher;
    public int PageNumber;
    public string ImageUrl;

    public Book(int BookID, string Title, string Author,
                string Publisher, int PageNumber, string ImageUrl)
    {
        this.BookID = BookID;
        this.Title = Title;
        this.Author = Author;
        this.Publisher = Publisher;
        this.PageNumber = PageNumber;
        this.ImageUrl = ImageUrl;
    }
}
```

Design of Default.aspx

When loaded, **Default.aspx** page should check whether the user is logged in or not. This check should be accomplished by a **cookie** check. The cookie should contain only the **first name** and **last name** of

the user. If no such cookie is defined, then **Default.aspx** should display a **message** and a **link** to the **Login.aspx** as in Fig. 1.

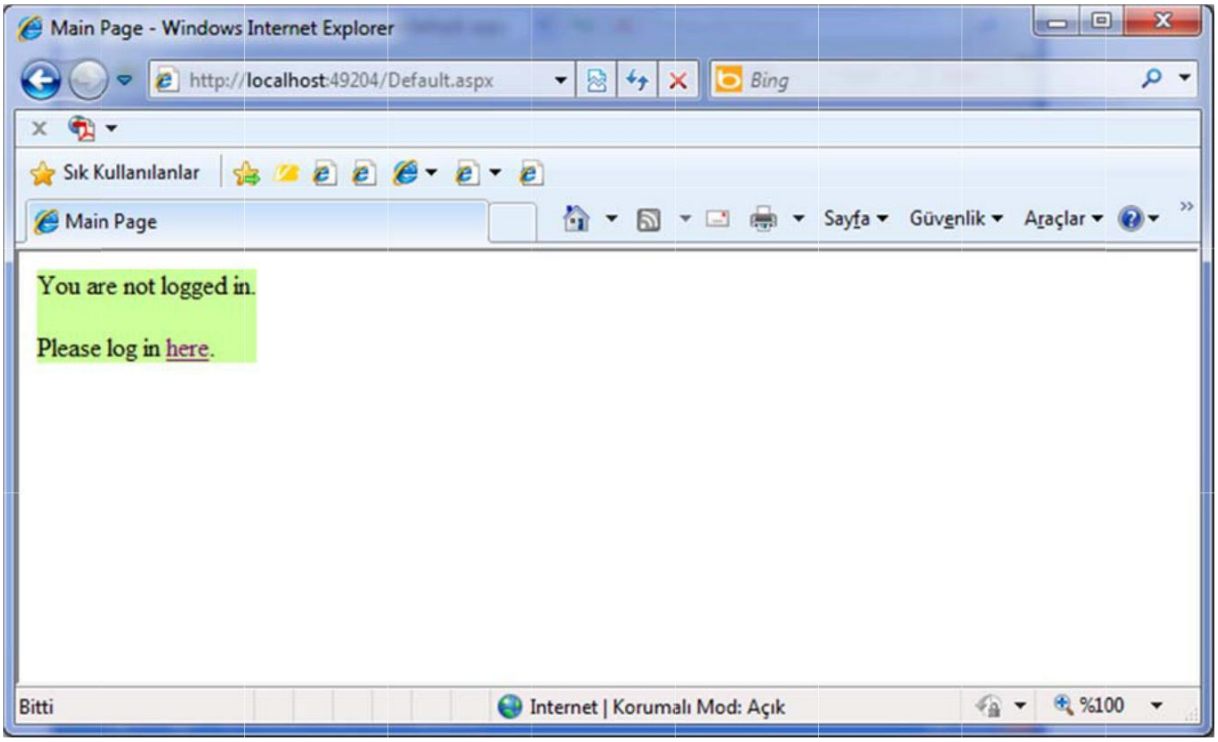

Fig. 1 – Default.aspx when user is not logged in.

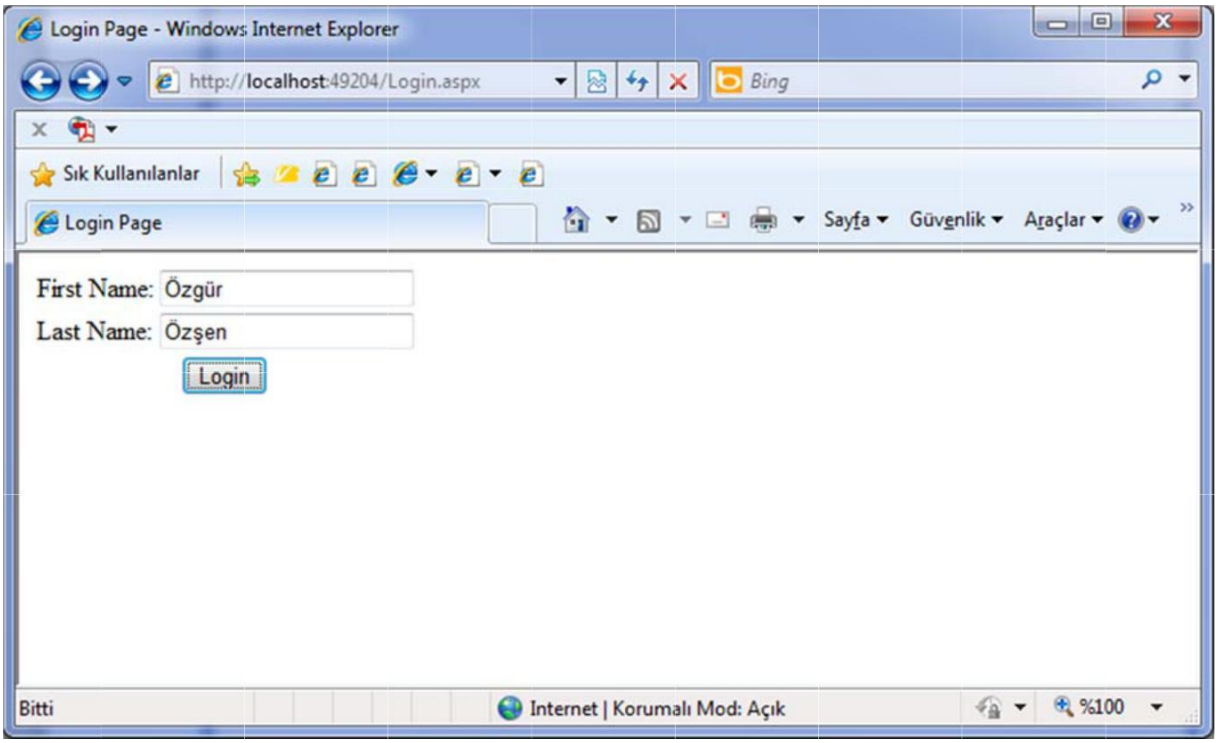

Fig. 2 – Login.aspx page

Design of Login.aspx

Login.aspx is very simple web page; it contains only **two textboxes** for **first name** and **last name**, and a **Login** button as in Fig. 2. There will be no password check. Any user that completes this form will be considered as a logged in user.

When the **Login** button is clicked , **first name** and **last name** will be written into a cookie named **UserInfo** for **one month** and the page will be redirected to **Default.aspx** and **Default.aspx** will display **user info** on the left and a **list of books** on the right as in Fig. 3. The information about the books is as the following:

Book ID	Title	Author	Publisher	Page Number
1	ASP.NET 3.5 Unleashed	Stephen Walther	Sams	1920
2	ASP.NET Ev olution	Dan Kent	Sams	384
3	Mastering Web Develop ment with Microsoft Visual Studio 2005	John Pa ul Mueller	Sams	848
4	Beginning A SP.NET 2.0	Chris Hart, John Kau fman, Dave Su ssman, and Chris Ullman	Wrox	792
5	Beginning A SP.NET 3.5 in C# 2008: From Novice to Professional, Secon d Edition	Matthe w MacDonal d	Apress	954

Table 1 – Book info

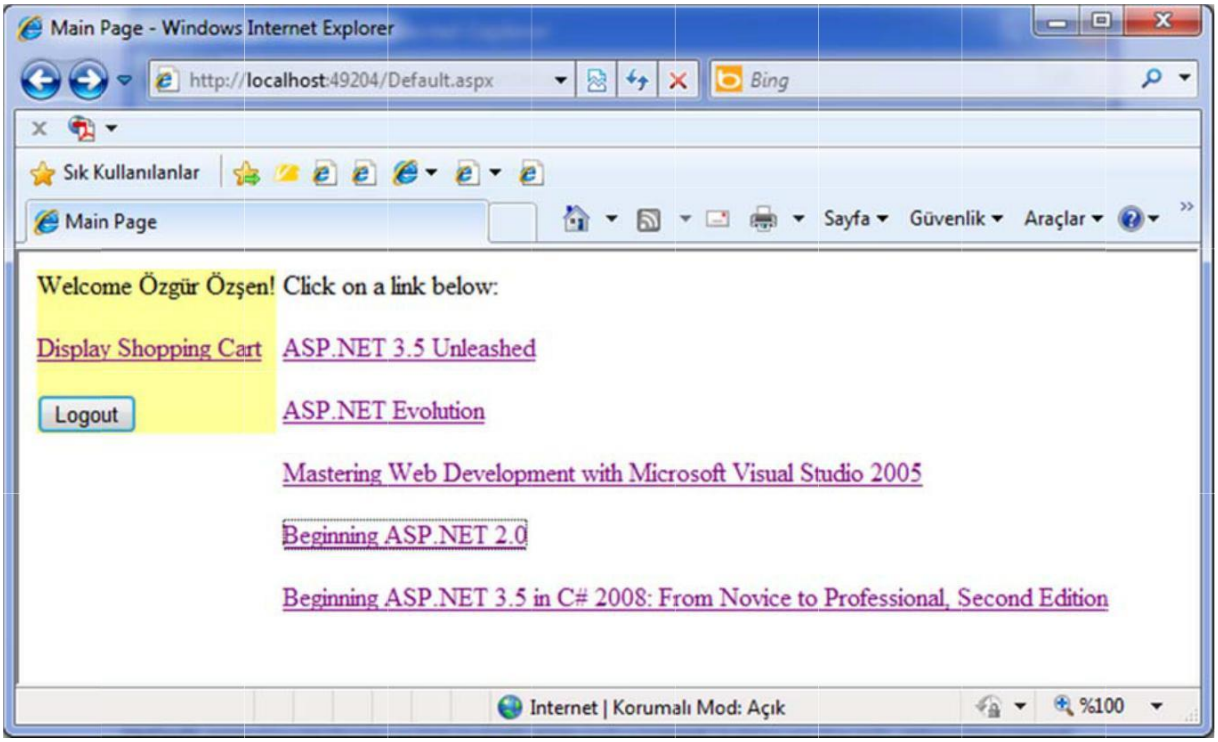

Fig. 3 – De fault.aspx page after user is logged in

Default.aspx should display a link to **Cart.aspx** and a **Logout** button on the left. When the **Logout** button is clicked, the **UserInfo** cookie should be remove d and Fig. 1 should be di splayed.

Default.aspx should display the titles of the books on the right with a **link** on each one. Each link should navigate to **BookInfo.aspx** with query string named **id** (e.g. **BookInfo.aspx?id=5**).

When **Default.aspx** page is first loaded, it should create five **Book** objects and put all them in to the **Session state**.

Design of BookInfo.aspx

When a book title is clicked in **D efault.aspx**, the information about the book should be displayed in **BookInfo.aspx** page as in Fig. 4. Book ID should be taken from **query string** with name **id**.

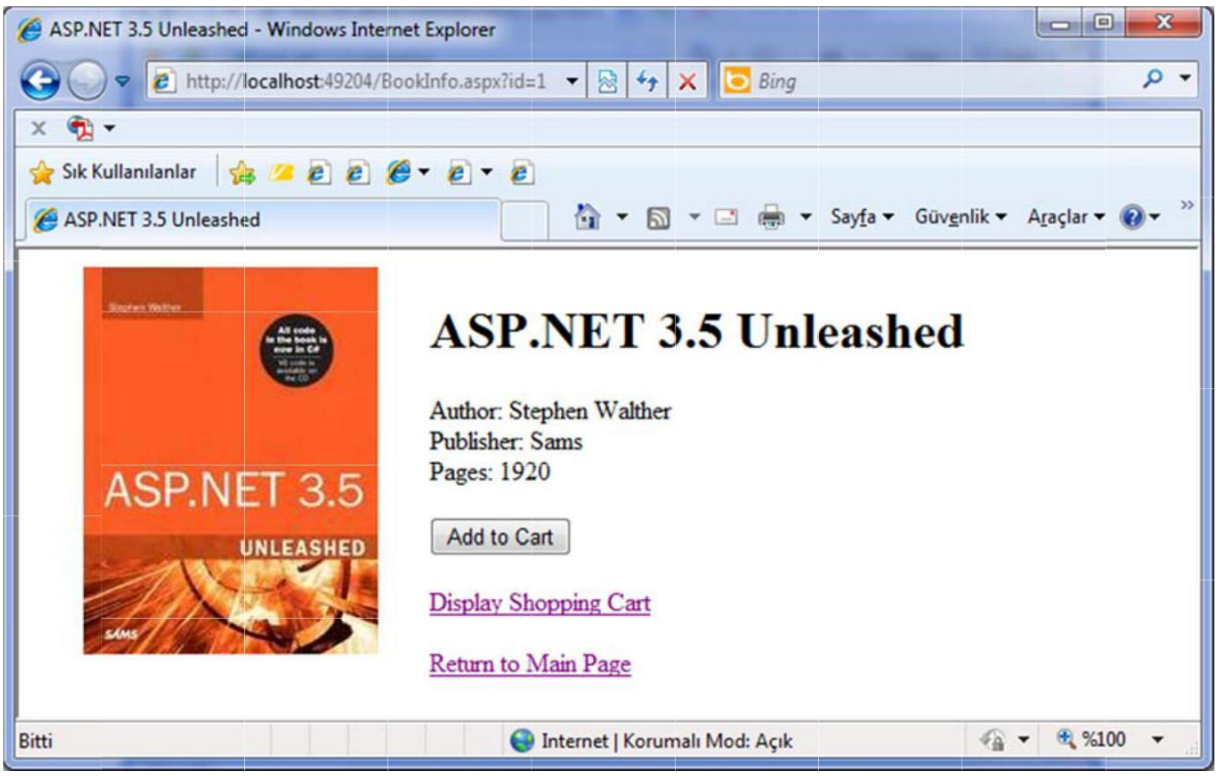

Fig. 4 – Bo okInfo.aspx p age with a vali d book id

If **BookInfo.aspx** is requested without any query string or the book ID does not exist, then it should display an appropriate message as in Figures 5 and 6.

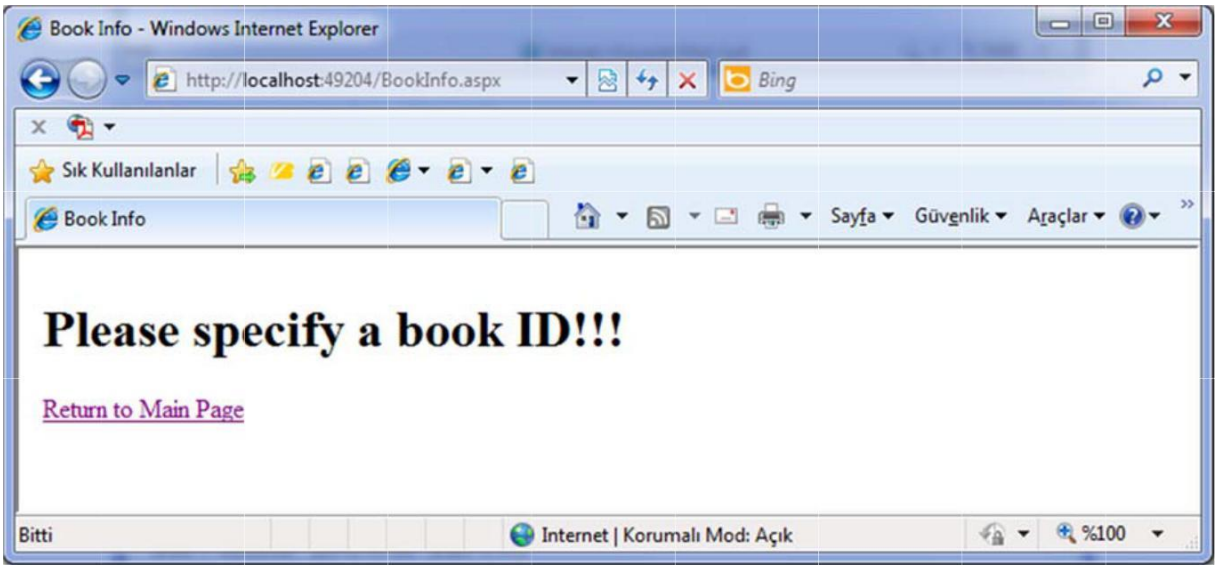

Fig. 5 – BookInfo.aspx when no query string is provided

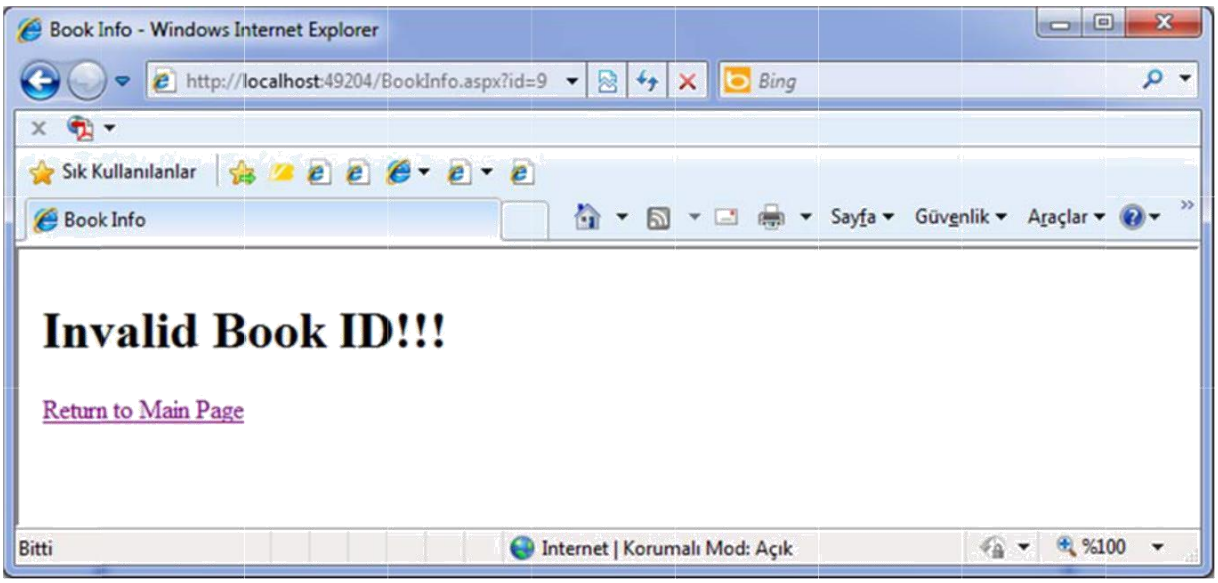

Fig. 6 – BookInfo.aspx when an invalid ID is provided

If a valid book ID is p resented, then **BookInfo.aspx** page should bring the desired book info from the **Session** state and di splay the book info with its cover image.

There also should be an **Add to Cart** button, a link to **Cart.aspx** and a link to **Default.aspx** as in Fig. 4.

When **Add to Cart** button is clicked, the book ID should be added in an **ArrayList** object in **Session** state that holds the selected book IDs. After the book is added into this **ArrayList** object, the user should be informed about this as in Fig. 7.

If that book is already in the cart, an appropriate message should be displayed as in Fig. 8.

BookInfo.aspx should also present links to **Cart.aspx** and **Default.aspx**.

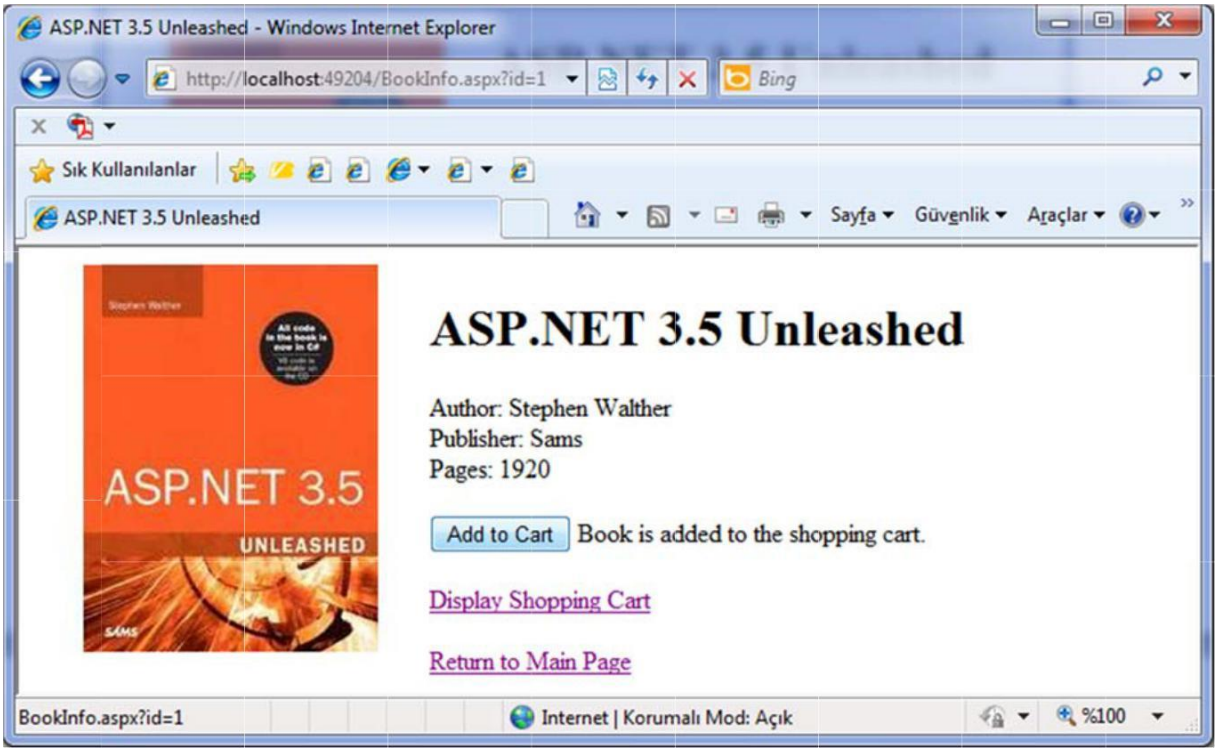

Fig. 7 – Bo okInfo.aspx after Add to Cart button is clicked and the book is successfully added to the cart

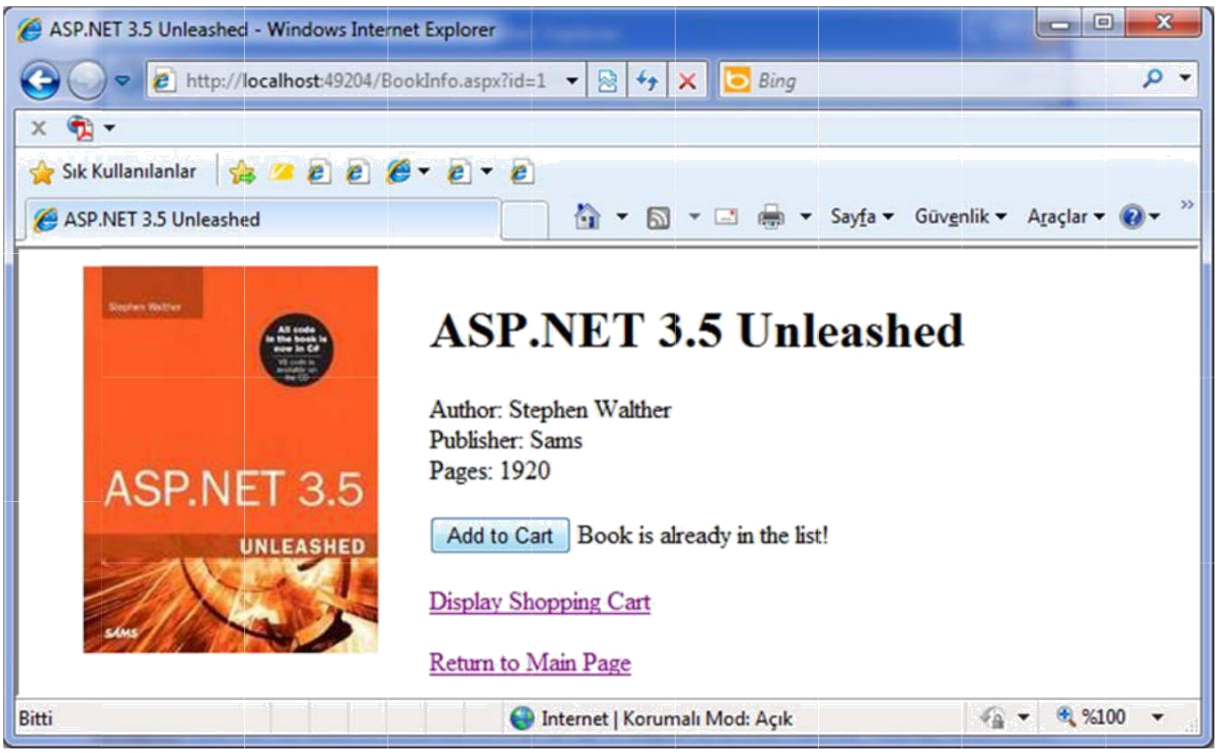

Fig. 8 – Bo okInfo.aspx page when Add to Cart button is clicked but the book was already in the cart.

Basic Information About *ArrayLis t* Class

ArrayList class can be used to st ore items in an array in object-oriented manner. You can store book IDs in an **ArrayList** object. An **ArrayList** objec t is created by the new operator:

```
ArrayList SelectedBookIndices = new ArrayList();
```

An item can be adde d to the **ArrayList** by **Add()** method:

```
SelectedBookIndices.Add(id);
```

Here, **id** can be of an y class. For the demonstration, we assume that **id** is of type **string**.

In order to check whether an item exists in the **ArrayList** or not, **Contains()** method can be used:

```
if (SelectedBookIndices.Contains(id))
{
    ...
}
```

To remove an item, **Remove()** method can be used. If you want to remove all items, you can use **Clear()** method.

Design of Cart.aspx

Cart.aspx should display all book s that are added to the shopping cart as in Fig. 9 . The IDs of the selected books should be taken from **Session** state as described above and information about the books should also be taken from the **Session** state which is put into the **Session** state by **Def ault.aspx** when it is first displayed. If there is no book in the shopping cart, a message should be displayed as in Fig. 10.

Cart.aspx page should also presents a link to the main page.

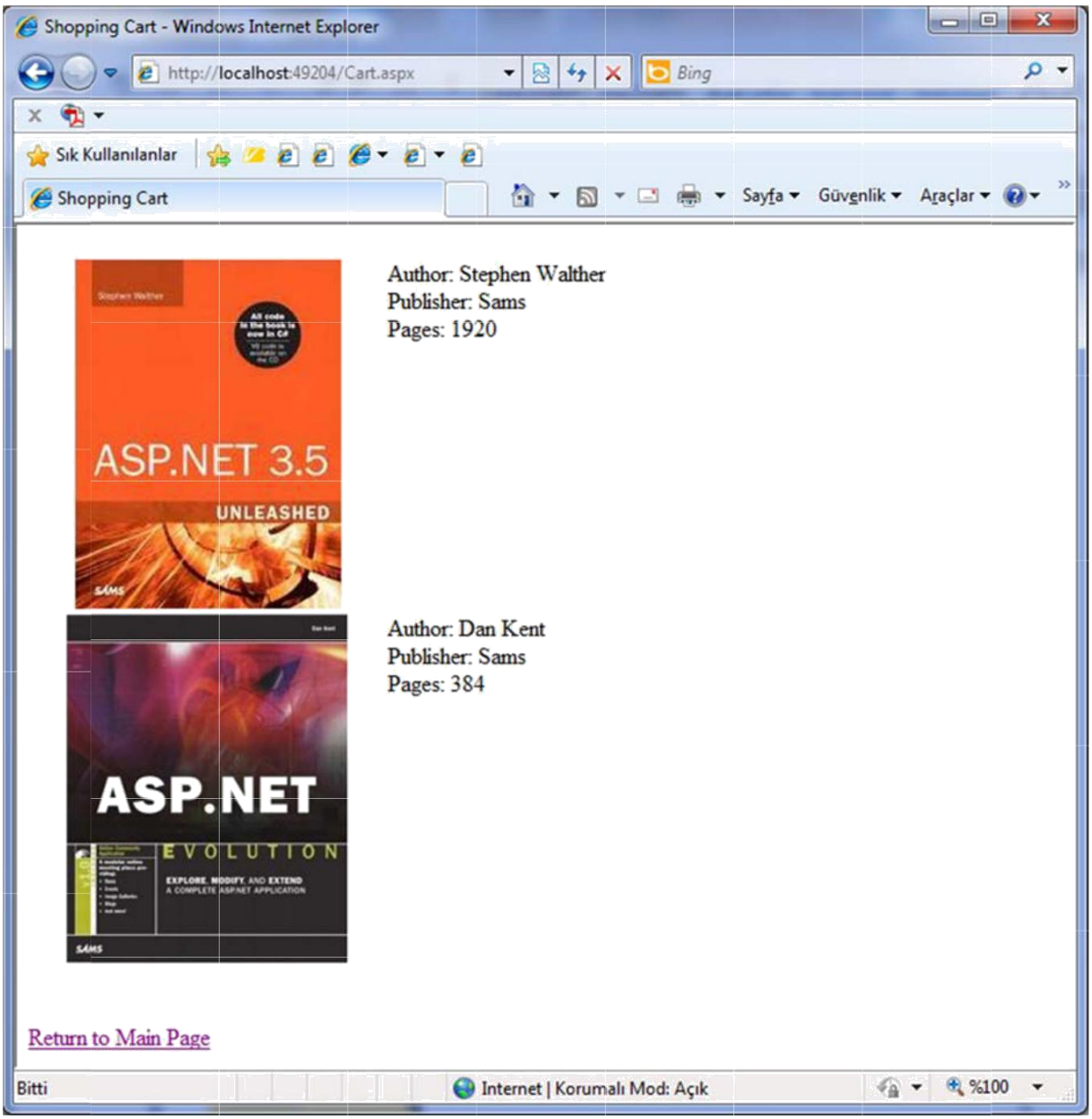

Fig. 9 – Cart.aspx when there are books in the shopping cart

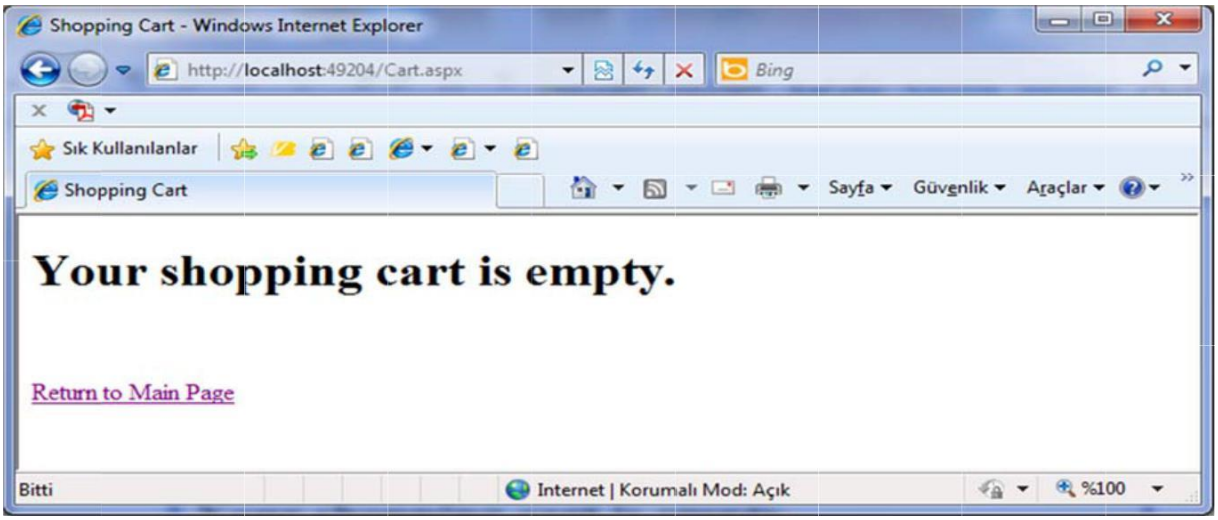

Fig. 10 – C art.aspx when the shopping cart is empty

Submitting the homework

Examle; Name-Surname-HW#.Zip