Server-Side Scripting: ASP.NET
Project 8: Final
200 points

NORTHEAST
WI Technical College

Create 5-page minimum database-driven website. A defined project is provided, but each student can create an ASP.NET final project on any topic, if approved by the instructor. The final project will not be accepted late.

All ASP.NET Final Projects must include the following requirements:

Req #	Required Elements	Possible Points
1	At least 5 pages	20
2	1 Master Layout to serve as the template	10
3	At least 1 Entity Framework Data Model used throughout app	20
4	At least 1 user input page with validation	20
5	At least 3 pages with Filtered Data via LINQ	30
6	At least 1 custom Data Model	20
7	At least 1 AJAX-enabled page	20
8	At least 10 HTML Helpers	20
9	At least 2 Partial Views	20
10	At least 2 Controllers with the same ActionResult	20
	Total Points	**200**

Henry's Bookstore

Students may elect to complete a pre-defined database-driven website based on a fictional business, Henry's Bookstore. The database for this project is on bitweb3.nwtc.edu. Your asp ID has read-only access to the _HENRY_ASPNET database. The details of this project are as follows.

Requirement #1: At least 5 pages

Henry's Bookstore internal Inventory Look-up system consists of 6 pages:
- *Home Page*
- *Browse by Author*
- *Book Details*
- *Browse by Location*
- *Browse by Publisher*
- *Contact Management*

Requirement #2: 1 master layout to serve as the template

Create the template for the site and apply CSS.

Requirement #3: At least 1 Entity Framework Data Model used throughout app

Connecting to your assigned database on bitweb3, use the Henry-related database to create a data model to use throughout the application.

Requirement #4: At least 1 user input page with validation

The Contact Management page contains validation.

Requirement #5: At least 3 pages with Filtered Data via LINQ
On the Browse by Author, Browse by Location and Browse by Publisher all filter data using LINQ

Requirement #6: At least 1 custom Data Model
There are opportunities for you to create a custom data model to optimally populate dropdowns.

Requirement #7: At least 1 AJAX-enabled page
Once the user submits valid data on the Contact Management page, a thank you message should appear using AJAX.

Requirement #8: At least 10 HTML Helpers
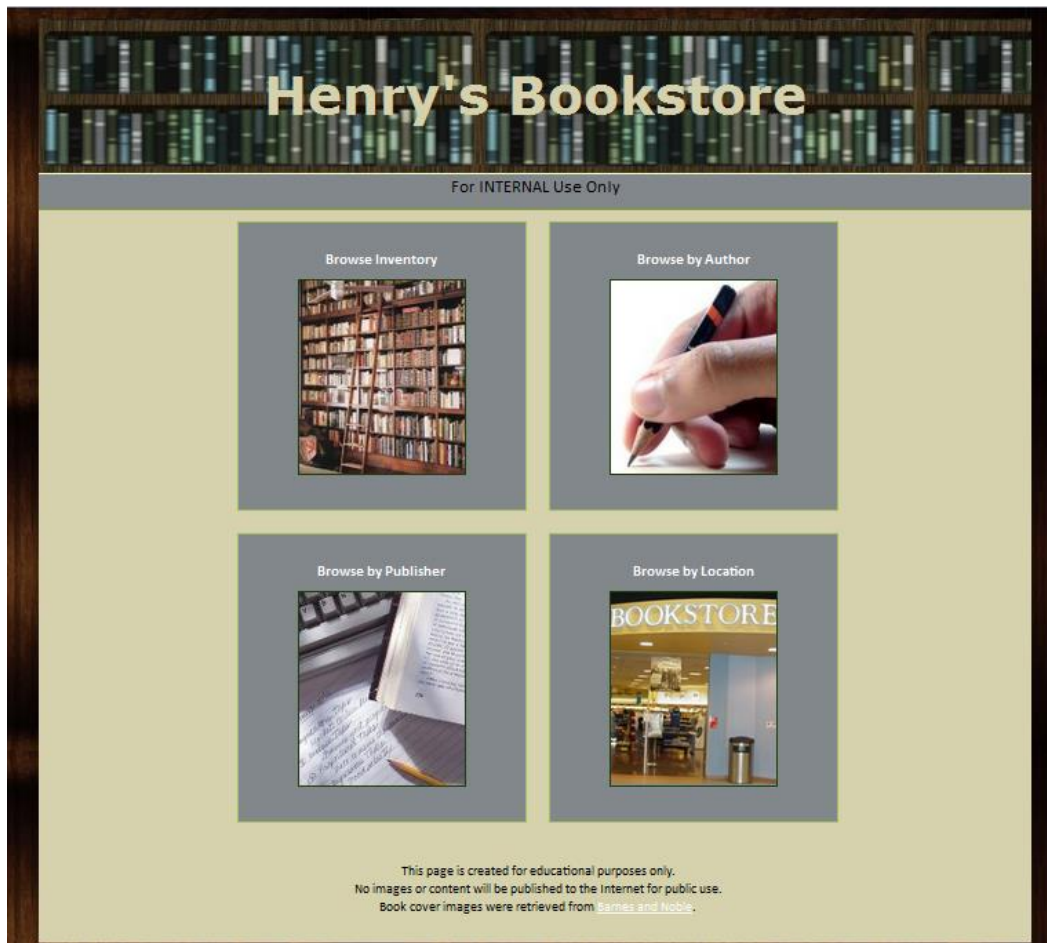There are numerous opportunities within this application to use HTML Helpers. Use at least 10.

Requirement #9: At least 2 partial views
At least 2 partial views can be used to support the data needs of this application.
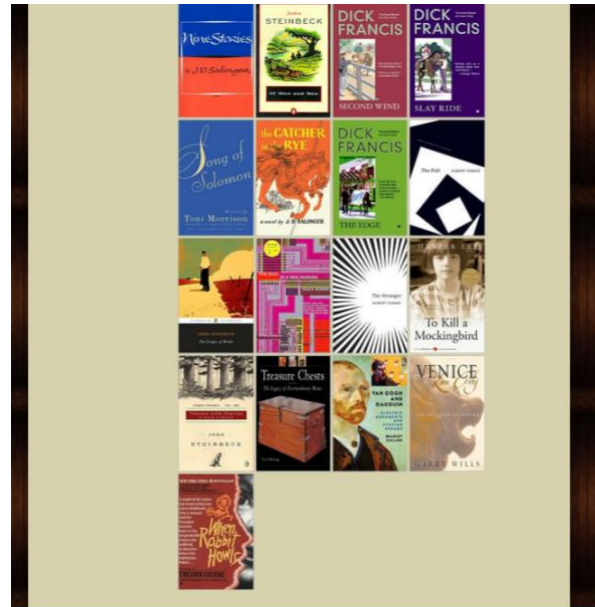
Requirement #10: At least 2 Controllers with the same ActionResult
There will be optimal times where using the same ActionResult, but one with HttpPost, will be necessary.
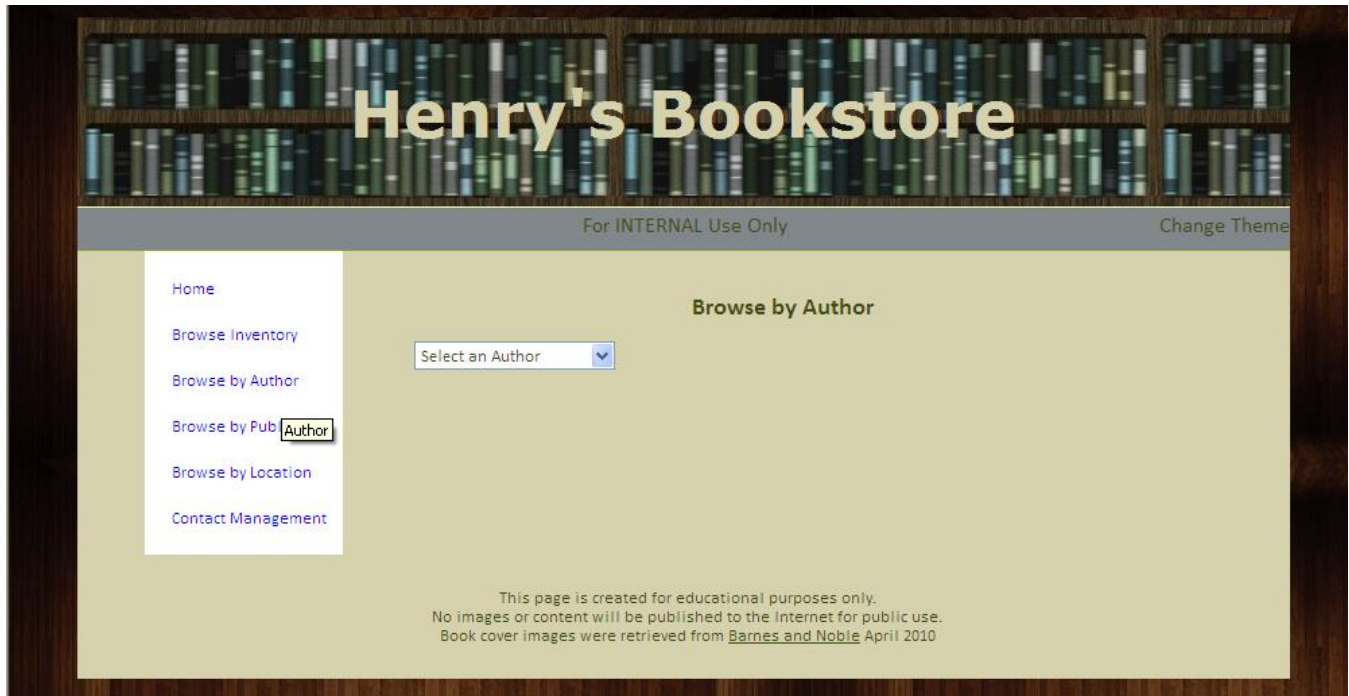
HENRY'S BOOKSTORE HOME PAGE

BROWSE BY INVENTORY

Each book in the BOOK table should be displayed, along with its thumb.jpg image.

BOOK DETAILS

Once a book is selected, its details page should display.

- The Publisher hyperlink goes to the Browse by Publisher page, passing in an id of the selected publisher (review routing assignment).
- The Author hyperlink(s) goes to the Browse by Author page, passing in an id of the selected author (review routing assignment).
- The Location hyperlink(s) goes to Browse by Location page, passing in an id of the selected location (review routing assignment).
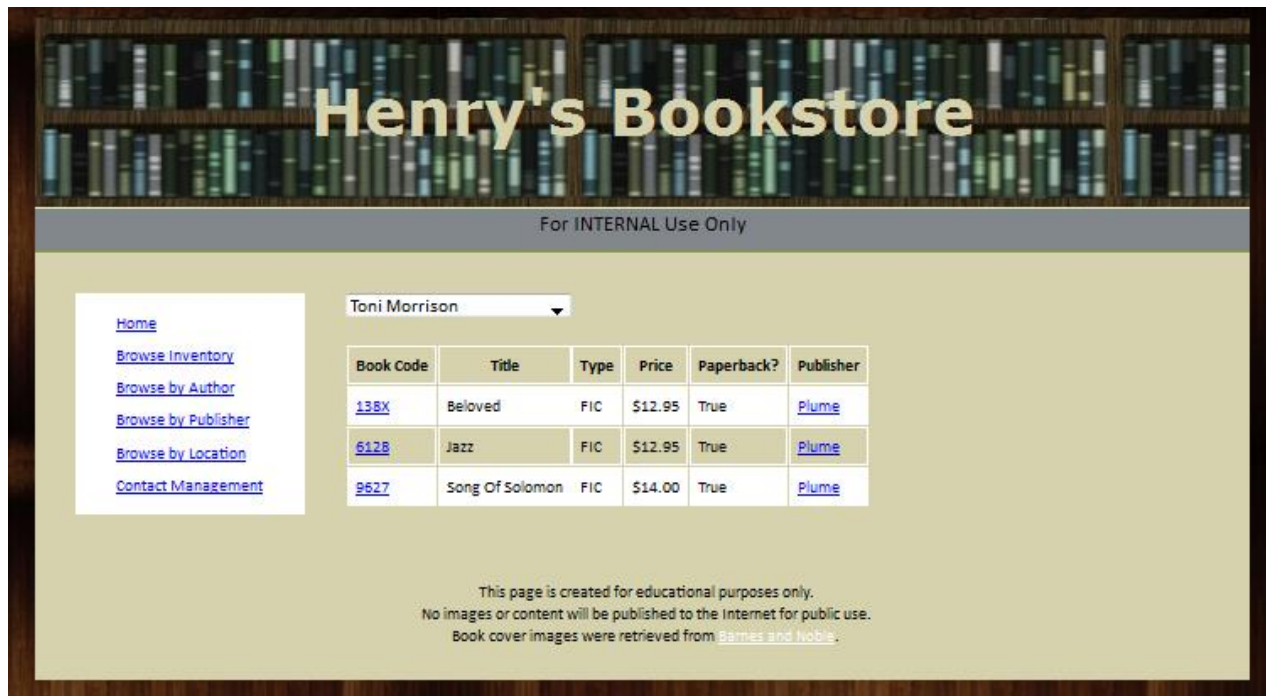
BROWSE BY AUTHOR

From the left navigation, Browse By Author has no selected author. In that case, a page with only a dropdown of all the Authors, alphabetized by Last Name, is provided.

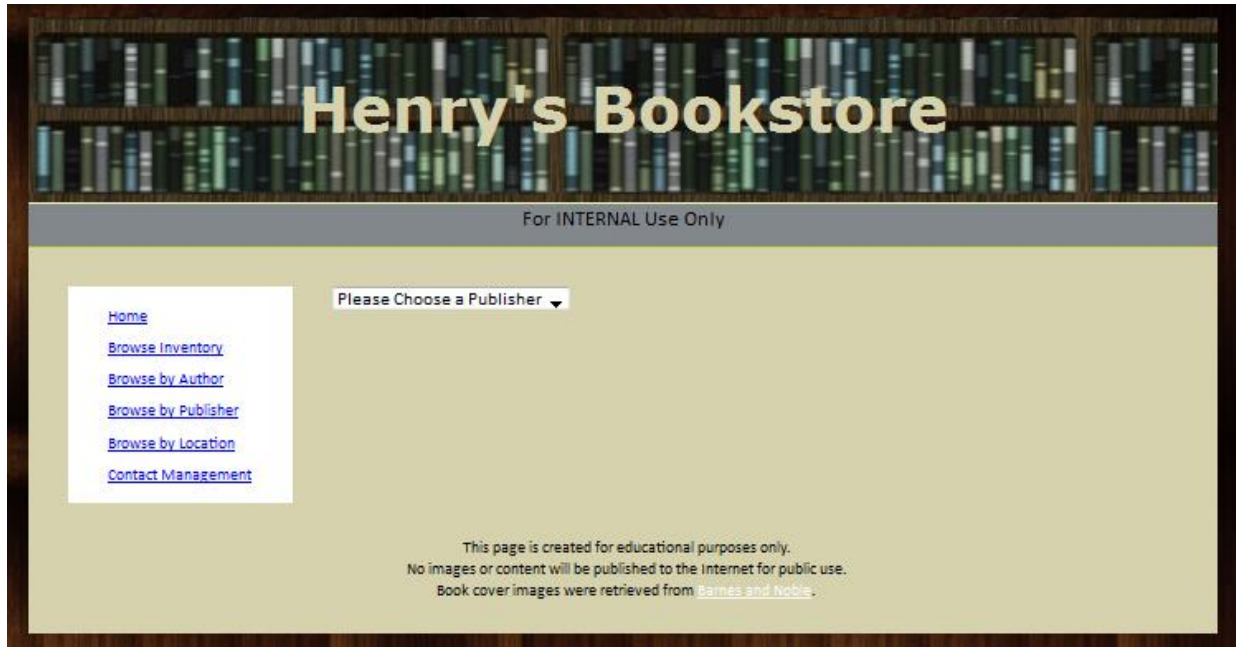As soon as a choice is made from the dropdown (onChange), the book(s) the author has written should display.

- The Book Code hyperlink goes to the Book Details page for that specific book.
- The Publisher hyperlink goes to the Browse by Publisher page, passing in an id of the selected publisher.
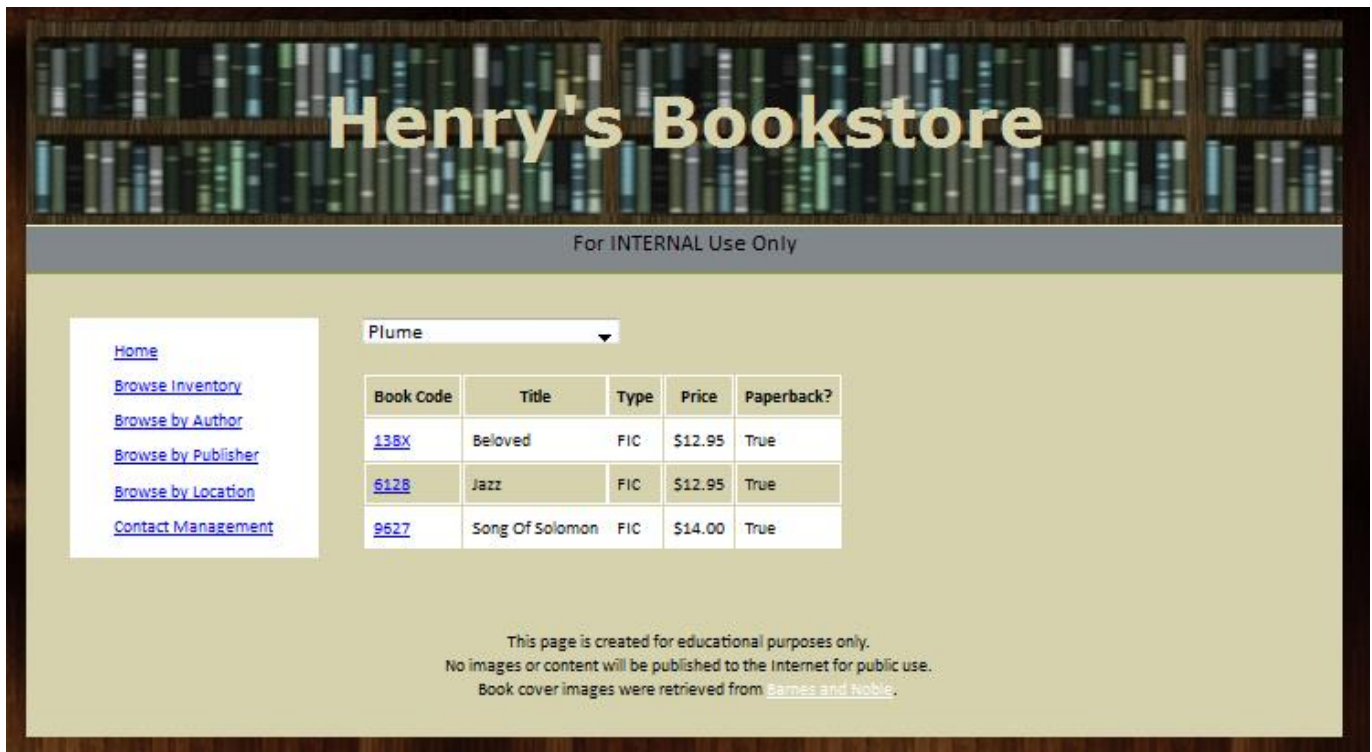
BROWSE BY PUBLISHER

From the left navigation, Browse By Publisher has no selected publisher. In that case, a page with only a dropdown of all the publishers, alphabetized by Publisher Name, is provided.
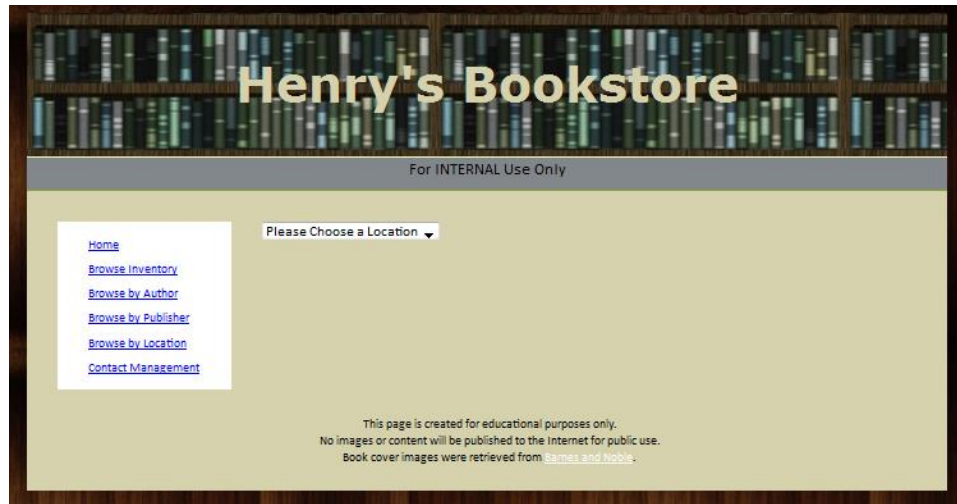

As soon as a choice is made from the dropdown (onChange), the book(s) the publisher has published should display.

- The Book Code hyperlink goes to the Book Details page for that specific book.

BROWSE BY LOCATION

From the left navigation, Browse By Location has no selected branch/location. In that case, a page with only a dropdown of all the branches/locations, alphabetized by Branch Name, is provided.

As soon as a choice is made from the dropdown (onChange), the book(s) the location has in inventory should display.

- The Book title hyperlink goes to the Book Details page for that specific book.

CONTACT MANAGEMENT

The Contact Management page should validate user information; all inputs are required. The Branch dropdown lists all branches/locations, alphabetically by Branch Name. The data is not saved to the database.

Once the user successfully enters valid data, a success message will display with AJAX.

Tips and Hints, if they're helpful!

<u>OPTIONAL INPUT PARAMETERS?</u>

Need to have an input parameter, but in some situations it may be null? Here's how you can change an ActionResult's input parameter as optional:

Integer optional input parameter example:

```csharp
public ActionResult Index(int id = -1)
{
        //somewhere in here, you will have to check if the id is greater than 0
    if (id > 0)
    {
       //a valid value was submitted
        }
         else
    {
       //a valid value was NOT submitted
        }
        return…
}
```

String optional input parameter example:

```csharp
public ActionResult Index(string id = "")
{
        //somewhere in here, you will have to check if the id is not null
    if (String.IsNullOrEmpty(id))
    {
       //a valid value was NOT submitted
        }
         else
    {
       //a valid value was submitted
        }
        return…
}
```

<u>NULLABLE MODEL IN VIEW?</u>

You may find it's possible a view has valid data in its model, but other times the model may be null. Here's how you can see if there's data in a model:

```csharp
@if (Model != null)
{
        //data populates the Model
}
```

GET THE "ONE" SIDE IN A "MANY" MODEL

A view may need to use a "MANY" side of a relationship as the model. Normally, you would use a loop to go through the "many" records. However, you will need "master" level data, or the "one" side of the relationship. You can do this by grabbing only the first record from the Model:

```
@Model.First().INSERT_WHATEVER_FIELD/PROPERTY_NAME_YOU NEED
```

DROPDOWN ON CHANGE EVENT

Part of the Browse by Author, Publisher and Location views will require something happen when a user chooses an Author, Publisher or Location. That event is an OnChange event. As part of the overload of an HTML Helper to render a dropdown, is a last property that could list the following code to automatically submit the form that contains the dropdown:

```
new{ onchange="this.form.submit();"}
```

BOOK IMAGE SIZES

The images used on the site have not been resized to fit specifically to this website. Force the following sizes for each of the book covers in your CSS:

Image Path	Image Size used on Website
[BOOK CODE]/thumb.jpg	125x193
[BOOK CODE]/large.jpg	397x600

HENRY_ASPNET Database Diagram