CIS 3342 Project 2 - Bookstore

The goal of this assignment is to give you some experience creating web applications with a GridView control and using dynamic binding.

Requirements:

1. Create a table named “Books” in your database with these fields

|  |  |
| --- | --- |
| **Field** | **Data Type** |
| ISBN | varchar(20) |
| Title | varchar |
| Authors | varchar |
| BasePrice | float |
| TotalSales | float |
| TotalQuantityRented | int |
| TotalQuantitySold | int |

1. Design the user interface (input form)
   1. Add textboxes for the user to enter their information StudentID, name, address, and phone number.
   2. Add a control for selecting a campus (Main, TUCC, Ambler, Tokyo, Rome, etc…).
   3. Put a **GridView** control on your form that will be used to order books. This input GridView display will be dynamic with columns. Each row in the GridView will correspond to a Book stored in the database.
      1. First column contains a checkbox to select a Book for purchase.
      2. Second column displays the Title value coming from the database.
      3. Third column displays the Authors coming from the database
      4. Fourth column displays the ISBN coming from the database
      5. Fifth column displays a drop-down list with the choice of hardcover, paper-back, or e-book.
      6. Sixth column displays a control for select either rent or buy.
      7. The last column displays a textbox for the quantity to order.
2. Use input validation
   1. The StudentID, name, address, and phone number textboxes cannot be blank.
   2. The user must select a campus.
   3. The user must select at least one Book before completing the order. This means they need to add a checkmark to at least one row in the GridView for a Book.
   4. The user must enter a quantity, a selection for a type of book, and a selection for rent/buy for every Book that is check-marked in the GridView.
3. Server-side processing
   1. Create a separate ClassLibrary project that contains the necessary classes to implement Component-Based Design. Name the library project **BookLibrary**
   2. The first class should contain class properties that will be displayed in the output GridView (in Part 5). The properties pertain to a single book that is ordered. Hint: in the CodeBehind of the order page, create an Arraylist of Books ordered where each Arraylist element is a Book object that was ordered.
      1. Title, Authors, ISBN, BookType, Quantity, Price, and TotalCost.
   3. Additional classes are used to model other entities needed in this software solution and they should contain a set of methods used in processing the order:

1. Display order output
   1. Display the StudentID, name, address, phone number, and campus.
   2. Display a dynamic 7-column GridView as Illustrated below. It needs to display the books that were ordered. The Arraylist of ordered books from part 4 should be bound to this GridView, and used to display the order.
   3. Column 7 - “Total Cost”, is calculated as Quantity x Price.
   4. The last row of the GridView should display the totals for the Quantity column, and the Total Cost column. You need to implement this as a footer of the GridView.

Output GridView

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Title** | **ISBN** | **Type** | **Buy/Rent** | **Price** | **Quantity** | **Total Cost** |
| … | … | … | … | … | … | … |
| … | … | … | … | … | … | … |
| **Totals** |  |  |  |  | … | $ … |

1. Management report
   1. Create a management report that will display the TotalSales, TotalQuantityRented, and TotalQuantitySold for each Book in the database.
   2. The data should be displayed in descending order.
2. Visibility control
   1. Control the visibility of both the input GridView and the output GridView displays. You want to show the input GridView when making an order and hide the output Gridview. Then, you want to show the output GridView and hide the input GridView after the order was processed.
3. Make the site professional and attractive. You do not need to write the CSS to style your site, but I expect your site to look professional. Make of colors, styles, images, alignments and proper presentation of the content. You may use CSS tools like BootStrap to design your site.
4. Good Design:
   1. Use server-side input validation where it’s necessary.
   2. Implement exception handling, so your programs don’t crash for any reason.
   3. Provide a consistent and logical navigation system. The user should never have to use the browser’s Back and Forward buttons to move between pages.
   4. Make your presentation clear to the user, providing on-screen instructions wherever needed both for data entry and error correction. If required data is omitted or entries are incorrect, the user should not have to re-enter data that is already correct.
   5. Create a good data model and implement the data model by creating the necessary tables in the database. You will be graded on the implementation of your data model.
   6. You need to use a proper naming convention for all controls and in your code. I expect you to properly name your classes, variables, functions, etc…
   7. **You must use component-based software design. This means writing as much code in classes and functions of classes instead of in the GUI.**

**Submission:**You need to publish your web application project to the cis-iis2 web server, upload your code to Blackboard, and provide the URL to your web application’s start page (Home Page / Table of Contents Page). Your start page should include a link to your project’s web application. Make sure a current version of your solution is located in your G:\cis3342 folder. Projects that are not submitted properly will not be graded.   
  
The grade for the required elements will be based on the published version of your web application. This means the elements must work from the published web application, not just the solution submitted through Blackboard. It’s important to make sure your published web application is the current version and that everything works. You shouldn’t publish to the Project folder after the project has been submitted. Otherwise, it will be considered late or may not be accepted if it’s after the deadline. If you make changes to your project, you will need to republish and resubmit the project. Generally, you shouldn’t publish or make changes until after you receive a grade.   
  
You need to zip the root folder for your solution into a single zip file and submit the assignment in Blackboard. To submit the assignment, you need to click the Assignment’s Title “**Project 2**” to view the submission form and upload the file.

**Make sure you properly submit your assignment and that it works. Programs that don’t run or don’t contain all the necessary files will not be graded and marked late.**