Project 2 – Online Music Store

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

Requirements:

1. Create a table named “Songs” in your database with the following fields.

Field	Data Type
SongID	int/varchar
SongTitle	varchar
MusicArtist	varchar
Genre	varchar
TotalPurchases	int

1. Create a table named “PurchaseTypes” in your database with the following fields.

Field	Data Type
MonthlyTotal	int
OneTimePurchaseTotal	int
AnnualTotal	int

2. Design the user interface (input form)
    a. Add textboxes for the user to enter their information name, address, and phone number, and credit card info (number, expiration month and year).
    b. Add a control for selecting either monthly auto-pay ($20), annual ($200) service, or one-time charge (cost of the songs they selected to buy).
    c. Put a **GridView** control on your form that will be used to choose songs to purchase. This input GridView display will be dynamic and contain the following columns. Each row in the GridView will correspond to a song stored in the database.
        i. First column contains a checkbox to select a SongTitle to purchase.
        ii. Second column is filled with the SongTitle value coming from the database.
        iii. Third column contains the music artist.
        iv. Fourth column contains a drop-down box used to select the song’s format (vinyl record, digital download, stream-only)
        v. Fifth column contains a textbox to enter the quantity to order.

3. Use input validation
    a. The name, address, phone number, and credit card info (number, expiration month and year) controls cannot be blank.
    b. The user must select an option to either for monthly auto-pay, annual, or one-time charge.
    c. The user must select at least one song before completing the order. This means they need to add a checkmark to at least one row in the GridView for a song.
    d. A quantity must be entered for every Song that is selected (check-marked) in the GridView.

4. Server-side processing
    a. Create a separate ClassLibrary project that contains 2 classes. Name the library project **MusicStoreLibrary**
    b. The first class should contain elements, as class properties, to be displayed in the output GridView (in Part 5). The elements that pertain to a single song that is ordered. Hint: in the CodeBehind of the order page, create an arraylist of songs ordered where each arraylist element is a Song object that was ordered.
        i. SongID, SongTitle, Quantity, Price, and TotalCost (based on the price of the song based on each format).
    c. The second class should contain the following methods/functions:
        i. One method receives the SongID and the format, and it computes the price. Each song should have its own price for each format.
        ii. One method that receives a SongID, order quantity, and format and updates the related total fields in the database to show the effect of placing the order.
        iii. Any additional methods you determine are necessary.

5. Display order output
    a. Display the name, address, phone number, credit-card info, and purchase type.
    b. Display a dynamic 5-column GridView as Illustrated below. It needs to display the songs that were ordered. The arraylist of ordered songs (objects made from the class in part 4b) should be bound to this GridView and used to display the order.
    c. Column 5 - "Total Cost", is calculated as Quantity x Price for the selected format.
    d. 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.
    e. Remember to insert the cost of the membership before the grand total in the footer.

Output GridView

SongTitle	Format	Quantity	Price	Total Cost
…	…	…	…	…
…	…	…	…	…
Totals		…		$ …

6. Song Statistics
    a. Add controls that allow the user to select what information is displayed. The user should be able to see the total sales by song.
        i. Use a GridView or Repeater to display the total sold for each song from the database.
        ii. The display should show the song title, artist, and genre.
        iii. The data should be displayed in descending order.
    b. Finally, allow the user to view the total songs sold by Genre.
        i. This can be done using SQL or it can be done programmatically using the data in the database to calculate the totals; you can decide how to accomplish this.
        ii. This information can be displayed using a GridView, Repeater, or simple label control.

7. Subscription Statistics
    a. Add controls that allow the user to view subscription information. The user should be able to see the total monthly, annual, or one-time subscriptions.
    b. The controls used to display this information should be hidden unless the user clicks a button to display this information.

8. Visibility control
    a. 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.
    b. Control the visibility of the GridViews used to display the statistics.

9. Good Design:
    a. Make the pages of your web application professional looking and presentable. This means they should make use of images, colors, and proper alignments to present the content. You're welcome to use tools like Bootstrap to enhance the presentation of your pages.
    b. Use server-side input validation where it's necessary.
    c. Implement exception handling, so your programs don't crash for any reason.
    d. 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.
    e. 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.
    f. 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.
    g. 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…
    h. **You must use component-based software design. This means writing as much code in classes and functions of classes instead of in the GUI.**