CIS 484 Spring 2022 Lab #2

This assignment is due by ***Sunday, February 27th by 12:00 NOON (Daytime).***

Requirements

This assignment builds on your Lab #1. Be sure to save/make a copy of your Lab1 Visual Studio solution folder (This will be your backup) in another location **before** starting this lab in case any questions come up with your Lab #1. You will use ASP.net, C#, and SQL Server again.

- **Your SQL Server Database for this lab is "Lab2".**
- **Your Solution/Project should be called "Lab2"**

For Lab2, you can once partner with TWO other people from your Section. Preferably, partner with the same person you worked with in Lab 1 to continue your work plus one more or you can complete the lab individually.

Requirements and Rubric:

No array of objects will be used anywhere in this assignment. You will not need to create instance objects of any class for this Lab. Data read/write will come strictly to/from the Database.	
Fulfilling the requirements listed here for Lab2 may require you to make changes to the structure and relationship of your database tables. You are free to change them as needed.	
You must have a simple Login with pages that are login-aware. The login must be functional – Users that are logged in stay logged in as they navigate pages/features of your site. Pre-Populate "existing" users in your DB (you will remove the user creation UI you implemented for Lab 1. Users cannot create their own accounts in this Lab).	3 Points ✓
The user must be able to Log out of the application. Some information pages should not be reachable if no one is logged in.	3 Points ✓
You must use a Master Page and sub-pages for your main site forms. Part of your master page's template contents must include some Login context (showing who is logged in while they are logged in).	3 Points ✓
No records will be deleted through the use of your UI	
When logged in, give the user the ability to edit their own information (name, etc.). See below	3 Points ✓
Ensure that Validator controls (the ones we discuss in class) are used on your WebControls to prevent user input errors.	3 Points ✗
Your database connection must be included and utilized in your Web.config file **only** (no connection strings in your actual ASP or C# Code.)	5 Points ✓
You must use both JOIN type query in your application in a sensible and value-added manner in at least one location.	5 Points ✓
Your code is functional and error free [Note: If it is not, the non-functional areas will impact points in other rubric items above!]	5 Points ✓
You have implemented all the Client Requirement Activities as noted below in the "Client Requirement Activities" section. Meaning: Users	20 Points

logging into your application can, depending on their roles, do all of the activities listed.	
Total:	**50 Points**

Client Requirement Activities

Depending on their roles, users logged into your application must be able to do the following:

Existing Requirements:

- ✓ Most, if not all, the tables from your ERD are implemented in your database. You can make changes to accommodate the requirements in this lab.
- ✓ Add, view, and edit individual students.
- ✓ Add, view, and edit individual DukeGroup members.
- ✓ Add, view, and edit individual Scholarship Opportunities.
- ✓ Add, view, and edit individual Internship Opportunities.
- Add, view, and edit individual Job Opportunities.

Does this count as our company or do we have to add specific job oppurtunities?

- ✓ User must be able to assign a DukeGroup member as a mentor to a student.
- ✓ Student must be able to apply for a scholarship, internship, or job opportunity. Application will be in this system and processed internally by the DukeGroup.
- A scholarship, internship, or job must be able to be awarded to a Student.

If we added a status to the application tables with pending, accepted, denied might satisfy this

New Requirements:

- ✓ Users cannot create their own accounts in this lab (no sign-up forms). Prepopulate your tables with at least 1-2 Members and 1-2 Students.
- ✓ Users can edit their own “account” info (name, email, etc. – Probably not their user name).
- ✓ There must be at least one value-added report that uses your JOIN query requirement. This could be a list of students, their mentor, and their internships awarded, if any, all displayed in one GridView or other format.

Quality of Code

I will judge your programs both by how well they work and based on the elegance of the code. The code should be well-documented and efficient. Proper indentation, descriptive variable names, and comments are essential. No extraneous variables or extra unnecessary lines of code. You must include your name and date at the top of all files (excluding .designer files) as a comment.

Deliverables

Provide a meaningful name to your files. When completed, you will zip your Web Site folder, along with test SQL Data, and submit that into Canvas by the deadline (this will include all of the files required including). Be sure to test all of your labs before you submit them to be graded in Canvas.