**Conestoga College - ACS & IT**

**Programming Microsoft Web Technologies – PROG2230**

**Assignment #5: ASP.NET Core: View Components, Authentication & Authorization**

**Due: Friday Dec 10, 2021 (by Midnight)**

**Total marks: 30 Worth: 6%**

**Introduction**

The goals of this assignment are to:

* Work through the steps of implementing authentication and authorization in a web app.
* Apply mechanisms to retract access to the app to authenticated users and certain functionality to users in an admin role.
* And develop a new “widget” for display on the app’s main page as a view component.

**The general idea**

~~This assignment is based on material from our~~ [~~Murach textbook~~](https://www.murach.com/shop-books/web-development-books/murach-s-asp-net-core-mvc-detail)~~, namely to add a view component, authentication, and authorization to their Quarterly Sales App. The starting code is available in the assignment folder.~~

**Please note**

~~Your solution~~ ***~~must only use techniques covered in class~~***~~. Also, I remind you to make sure you do your own work on this assignment and resist any urge to copy code from any other source - e.g. your classmates, the web, etc. Everyone’s solution will be run through~~ [~~Moss~~](https://theory.stanford.edu/~aiken/moss/) ~~to check for academic integrity violations. There is zero-tolerance for such violations and any encountered with be dealt with in accordance with~~ [~~Cone HYPERLINK "https://lib.conestogac.on.ca/academic-integrity/penalties"stoga’s policy~~](https://lib.conestogac.on.ca/academic-integrity/penalties).

**How will it be graded?**

~~Accompanying the exam will be an Excel marking sheet that details how your grade will be calculated so you will want to make sure that you are doing everything as it’s laid out there.~~

**What/how to submit?**

~~Zip up your entire solution into~~ ***~~one zip file~~*** ~~and submit that file to the eConestoga dropbox for the assignment.You can submit multiple solutions but only the latest (i.e. most recent) one will be looked at and evaluated.~~

**Part I: Add authentication & authorization to the Quarterly Sales app**

~~In this part, you’ll update the existing Quarterly Sales app (see starting code) to use authentication and authorization. This should be a fully functional app as is but to set it up do the following:~~

* ~~In the DB connection string in the~~ *~~appsettings.json~~* ~~file append your student number to the end of the database name.~~
* ~~Then run Update-Database, and test the app.~~

~~The following screenshots will help you understand what you need to build. First, the navbar for an anonymous user:~~



The navbar for an authenticated user not in the Admin role:



The navbar for an authenticated user in the Admin role:



The Manage Users page:



**Specifications:**

* The app should provide a way for users to register and log in.
* When registering, the user must provide a username and a password. Here’s the password criteria:
* Passwords must be at least 6 characters.
* Passwords must have at least one uppercase character (A-Z).
* Anonymous users can view the sales data on the main page but aren’t authorized to add sales or employees.
* Authenticated users are authorized to add sales, but only authenticated users in the Admin role are authorized to add employees.
* On startup, the app should create an Admin user named “*admin*” with a password of “*P@ssw0rd*”.
* For anonymous users, the navbar should provide a Register link and a “*Log In*” link.
* For authenticated users, the navbar should display the user’s name and provide a “*Log Out*” link. In addition, the navbar should display a link to the Add Sales page.
* For authenticated users in the Admin role, the navbar should display links to the Add Employee page and the Manage Users page.
* The Manage Users page should allow a user in the Admin role to add and delete users, and assign users to and remove them from the Admin role.

**~~Part II: Add TopQuartersByEmployee view component~~**

~~In this section you need to build and make use of a TopQuartersByEmployee view component that:~~

* Accepts a parameter that represents the number of quarters to display
* And then displays that number of the top quarters in terms of sales for a given employee

For example, if configured to show the top 3 quarters by employee, then your view component should look something like this and it means Katherine Johnson had the most sales in a quarter and Homer Simpson has the second most, and of course Marge had the third most:



When you incorporate the view component into the app, you should apply a little Bootstrap layout styling so that the view component appears to the right of the table of sales:

