



Assignment #2

School of Computing and Academic Studies

COMP4870

Programs: CST

Due: Tuesday Mar 16, 2020 at 11:59 pm

ASSIGNMENT DETAILS

The assignment involves development of a dockerized client-side blazor app that enables CRUD operations on *Speaker* data:

	Example
FirstName	Jim
LastName	Potter
Email	jim@potter.ca
Mobile Number	604-123-4567
Area of specialization	Cloud Architect
City of residence	Chilliwack
Province of residence	BC
Employer	BC Hydro

The purpose of this assignment is familiarity with Docker & Client-side Blazor.

- The Assignment will be done in teams of two students assigned by your lab instructor.
- The landing page of the client-side lists speakers as shown below:

Add Speaker

Jim Potter	Edit	Delete
Jane Douglas	Edit	Delete
Tom Gardner	Edit	Delete
Ann Lee	Edit	Delete
James Jones	Edit	Delete
Susan Taylor	Edit	Delete
Peter White	Edit	Delete
Philip Fox	Edit	Delete
Donna Ray	Edit	Delete

- The "Add Speaker" button allows you to add a speaker
- The *Edit* & *Delete* links allow you to update and remove items accordingly
- *Add*, *Edit* & *Delete* forms must be displayed on separate pages.
- Adequate validation exists on Add & Edit pages
- Use this version of SQL Server Docker Image: mcr.microsoft.com/mssql/server:2017-CU8-ubuntu
- Create a *docker-compose.yml* file that runs the application such that:
 - Data is saved in a SQL Server database in a separate container
 - The server-side API application runs in a container
 - The client-side application runs in a container
 - To run the application the marker simply runs "docker-compose up" and views the web page by going to <http://localhost:8888>

Container 1 (ASP.NET API)

Container 2 (DB)

Container 3 (Blazor)

SUBMISSION:

- Do not include unnecessary files that are not needed in your submission. Delete the *node_modules*, *.git*, *.vscode*, etc... folders before you ZIP your file to make it as small as possible.
- Put the following information into a *readme.txt* file:
 - your names, BCIT ID numbers and your preferred email addresses. Avoid your my.bcit.ca email account because it has file attachment restrictions. This is necessary in case the marker wishes to urgently contact you.
 - what you have not completed
 - any major challenges
 - any special instructions for testing your app
 - place the *readme.txt* file inside a *docs* folder in the root your web application
- Assignments must be zipped (.zip extension) and uploaded to the drop-box folder for *Assignment 2* in D2L (Learning Hub). Do not use any compression utility other than zip.
- Assignment Zip files should be named **LastName1_ LastName2_ass2.zip**. (Example: Doe_Smith_ass2.zip).
- Your ZIP file will include all directories and files comprising your entire web app.
- Each team must make only one submission. If you make more than one submission then you should use the same file name and clearly version the file by adding *_v2*, *_v3*, etc.
- As you submit your solution into D2L, add a D2L comment with the URL of your online deployed solution

COMP4870 ASSIGNMENT 2 MARKING GUIDE:

Task	Max Mark	Actual Mark
Landing page shows list of items with Edit & Delete links + Add button	4	
Able to Add, Edit & Delete data. Add, Edit & Delete forms displayed on separate pages.	7	
Docker <ul style="list-style-type: none">• API in container• DB in container• Client-side app in container• docker-compose up• app is accessible using http://localhost:8888	5	
Adequate validations added to Add & Edit pages	2	
Other <ul style="list-style-type: none">• student names on the home page• app & submission names as prescribed• followed instructions• Design and coding conventions• Contents & quality of <i>readme.txt</i> file	2	
TOTAL:	20	

NOTE: You will get a mark of zero if your docker-compose file does not work.