**Assignment 2 – Server-Side Scripting**

This assignment requires you to extend your ASP.NET Core Web Application from Assignment 1 and integrate it with a SQL Server database to perform CRUD operations. Your mark counts for 25% of your final grade.

Due Dates:

Part 1: **Sunday, Nov 15 @ 5:00 pm**

Part 2: **Sunday, Dec 6 @ 5:00 pm**

Part 3: **Sunday, Dec 13 @ 5:00 pm**

**Submission Requirements via Blackboard**

**Part 1:**

* Link to your private Web Site repository (invite me – username **ifotn** - as a collaborator). This can be the same repository you used for Assignment 1.
* Link to your Assignment site running on Azure (or alternative server that supports .NET)

**Part 2:**

* Resubmit the same repository and Azure links you submitted in Part 1 as these should now be updated with the additions for Part 2.

**Part 3:**

* Resubmit the same repository and Azure links – this time that points to the url containing your api (e.g. [http://[yoursubdomain].azurewebsites.net/api/[controllername](http://[yoursubdomain].azurewebsites.net/api/%5bcontrollername)] - you submitted in Part 1 as these should now be updated with the additions for Part 2.

**All work must be your own.**  Failure to submit an independent assignment will result in a grade of zero. If your database tables are exactly the same (or very close to the same) as another student’s, it will potentially be considered as Academic Misconduct. Code will be checked carefully with code comparison software. **You are not permitted to download our in-class application, simply modify the words then submit these back to me as your own work.**

**All submissions are subject to a Code Review where you may be asked to walk through and explain your work.**

# Description: Dynamic ASP.NET MVC Site – Version 2

You will enhance the application you built for Assignment 1 by adding:

* Authentication – both Local and Social
* Unit Testing
* Web API

**Application Requirements:**

**Part 1 (15 marks)**

1. Configure Identity in Startup.cs
   1. Disable Requiring Account Confirmation by Email
2. Create an account for me with these credentials:
   1. [rich@gc.ca](mailto:rich@gc.ca) / Test123$
3. Modify the site in the following ways:
   1. Make all Views where users can add, edit, or delete data PRIVATE, so only authenticated users can access them
   2. On your Index views, anonymous users can view the list of data but cannot see the Create, Edit, or Delete links
4. Enable Social Authentication with Google. Install the required NuGet Package. Create new keys for your assignment rather than using your existing keys from our in-class application. Authorize the *azurewebsites.net* domain and add BOTH your local and Azure Google signin redirect uri’s in the Google Console. Store all API Keys in either:
   1. Project Secret Manager AND Azure Configuration Settings (Recommended) – OR –
   2. appsettings.json file
5. Use GitHub to make regular, descriptively-named commits (minimum 4 over at least 2 days) and ensure you have a descriptive readme README.md file that explains the purpose of your application. Use .gitignore so the Packages folder does not get included in your online repository.
6. Publish to the site to Azure or any other web server that supports ASP.NET and SQL Server. Include this link in your README.md file.

**Part 2 (11 marks + 2 bonus)**

1. Create a Unit Testing project within your solution and create a Test Class for ONE of your Controllers.
2. Write Unit Tests for each method in ONE Controller to get as close as possible to 100% Code Coverage in this controller (roughly 23 tests), using Moq to create mock data. Use interfaces with your mock data and modify your controller so they use the mock data when testing but the live database in production.
3. Use GitHub to make regular commits (minimum 2) and create a README.md file that explains the purpose of your application. Use .gitignore so the   
   Packages folder does not get included in your online repository.

**Part 3 (11 marks + 2 bonus)**

1. In your project, add a new subfolder called **api**
2. In the new api folder, create **one** scaffolded Web API controllers with full Create-Read-Update-Delete functionality for **one** of the tables in your database
3. Use GitHub to make regular commits.
4. Publish to the site to Azure or any other web server that supports ASP.NET and SQL Server. Include this link in your README.md file.
5. Bonus #2: Add an advanced .NET Core Web API feature that we have not yet learned in class. You MUST document this feature in the README.md so I know what to look for. (Part 2)

**Evaluation Method**

Work is evaluated based on how your application performs on the following items:

# Evaluation Criteria

Part 1: 15 marks

Part 2: 11 marks

Part 3: 11 marks (2 bonus marks available)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **0-2** | **3-4** | **5-6** | **7-8** | **Marks** |
| **Authentication**  (Part 1) | No Authentication | Some implementation | Only minor omissions | Full, functional Auth Model with Working Local Registration & Login; Create, Edit, Delete for authenticated users only | **8** |
| Social Login (Part 1) | No Social Login | Partly correct but incomplete | 1 working provider | Google login fully functional | **4** |
| Unit Testing (Part 2) | No unit tests | Incomplete unit test implementation | Only minor changes outstanding | Functional Pass / Fail Tests for all Methods in **ONE** Controller using Mock Data | **8** |
| Web API (Part 3) | No API | API partly working | API working with minor errors or omissions | Complete Web API with working CRUD operations for one of your Controllers | **8** |
| Version Control (2 marks for each Part) | None | < 4 commits, incomplete README.md |  | Min 4 descriptively-named commits, full README.md | **6** |
| Server Deployment (1 mark for each Part) | No deployment | Deployed with errors |  | Full deployed online. Link in README.md | **3** |
| **\*\*\* Bonus 2\*\*\***  (Part 2) |  |  |  | Up to 2 bonus marks for any extra working API functionality of your choosing. Documented in README.md | **2** |