COMP229 – Web Application Development

Assignment 2
Express Portfolio – Authentication
Due Week #6 (October 23rd, 2020) @ midnight

Value 15%

Express Portfolio - Authentication   **Maximum Mark: 100**

Overview: Using the Portfolio Web Site you developed in Assignment 1, restructure the web site to include a secure section of you site that displays a list of contacts.

Instructions :
This Secure Area must include the following components – a user **Login View, a Business Contacts List View** and an **Update View.**

1. In addition to your site's main **Nav Bar** and **page template components**, your **Login View** should include the following components **(4 Marks: GUI, 25 Marks: Functionality):**
   a. A Form that includes a **username field**, a **password field** and a **Login Button.** (4 Marks: GUI, 4 Marks: Functionality).
   b. The application database will contain a **user collection.** The single document in the collection will include your **username**, **password, email address** and other relevant information. The username and password credentials will be used for authentication. No Registration form is required. (5 Marks: Functionality).
   c. If the user is authenticated, he will be taken to the **Business Contacts List View**, otherwise he will be **redirected** back to the Login View if his username and/or password is incorrect (8 Marks: Functionality).
   d. If a user attempts to access the secure area of your site, they should be redirected back to the Login View (8 Marks: Functionality).
2. Your **Business Contacts List View** should include the following components **(8 Marks: GUI, 16 Marks: Functionality):**
   a. An alphabetically **sorted** list of contacts should appear on this page. The connection to the database you created will display all the contact data in a table. The table will only include The **Contact Name**, **Contact Number** and **Email Address** (4 Marks: GUI, 8 Marks: Functionality).
   b. Include an **Update Button** and a **Delete Button** at the end of each Row of the Table. The Update button links to the **Update View.** The Delete Button **removes** the contact from the database (4 Marks: GUI, 8 Marks: Functionality).
3. The Update View will include the following components **(12 Marks: GUI, 12 Marks: Functionality):**
   a. A **Form** will allow the user to update any contact details (**Name**, **Contact Number**, **Email Address**, etc.) (4 Marks: GUI, 4 Marks: Functionality)

b. The Form will include a **Delete Button** that will remove the user from the database and return the user to the Contact List View (4 Marks: GUI, 4 Marks: Functionality).

c. The Form will include a **Cancel Button** that will return the user to the Contact List View (4 Marks: GUI, 4 Marks: Functionality).

4. Include **Internal Documentation** for your site **(5 Marks: Internal Documentation):**

a. Ensure you include a **comment header** for your **CSS and JavaScript files** that indicate: the **File name**, **Student's Name**, **StudentID, and Date,** (2 Marks: Internal Documentation).

b. Ensure you include a **section header** for all of your **HTML structure, CSS style sections,** and any **JavaScript functions** (1 Marks: Internal Documentation)

c. Ensure all your code uses **contextual variable names** that help make the files human-readable (1 Marks: Internal Documentation).

d. Ensure you include **inline comments** that describe your GUI Design and Functionality. **Note:** Please avoid "over-commenting" (1 Marks: Internal Documentation)

5. Share your files on **GitHub** to demonstrate Version Control Best Practices and push your site to a cloud host **(4 Marks: Version Control, 4 Marks: Cloud Hosting).**

a. Your repository must include **your code** and be well structured (2 Marks: Version Control).

b. Your repository must include **commits** that demonstrate the project being updated at different stages of development – each time a major change is implemented (2 Marks: Version Control).

c. You must deploy your site to your Cloud Server using **git** (4 Marks: Cloud Hosting).

6. Create a Short Video presentation on **YouTube** or another streaming provider. You must include a short **PowerPoint** (or Google Slides) Slide Deck that includes a **single slide** to start your video (10 Marks: Video)

a. The **first** (and only) **Slide** of your Slide Deck must include a **current image** of you (no avatars allowed) that is displayed appropriately on the page. You must also include your **Full Name**, **Student ID**, the **Course Code**, **Course Name,** and your **Assignment** information. (2 Marks: video)

b. You will **demonstrate** your site's functionality. You must show each page working properly (2 Marks: Video)

c. You will **describe** the code in your app.js file that drives the functionality of your site (2 Marks Video).

d. Sound for your Video must at an appropriate level so that your voice may be **clearly heard,** and your screen resolution should be set so that your code and site details are **clearly visible** (2 Marks: Video).

e. Your Short Video should run no more than 5 minutes (2 Marks: Video).

SUBMITTING YOUR WORK
Your submission should include:
1. A zip archive of your website's Project files
2. A link to GitHub
3. A link to your live site on a Cloud Host (or other hosting service)

Evaluation Criteria

Feature	Description	Marks
GUI / Interface Design	Display elements meet requirements. Appropriate spacing, graphics, colour, and typography used.	24
Functionality	Site deliverables are met and site functions are met. No errors, including submission of user inputs.	53
Internal Documentation	File header present, including site & student name & description. Functions and classes include headers describing functionality & scope. Inline comments and descriptive variable names included.	5
Version Control	GitHub commit history demonstrating regular updates.	4
Cloud Deployment	Deploy site to Cloud Service.	4
Video Presentation	Your short video must demonstrate your site and describe your code	10
Total		**100**

This assignment is weighted **15%** of your total mark for this course.

Late submissions:
- 20% deducted for each day late.

External code (e.g. from the internet or other sources) can be used for student submissions within the following parameters:
1. The code source (i.e. where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.