COMP125– Client-Side Web Development

Assignment 3
Mini Portfolio – JSON Data & AJAX Components

Due: Week 12 (Friday August 3rd, 2018) @ Midnight

Value 15%

Mini Portfolio - JSON Data & AJAX Components **Maximum Mark: 46**

Overview: Extend your mini portfolio from Assignment 1 and 2. You will write JavaScript code to read your paragraph data from a JSON file using AJAX and populate your page with this data. Ensure that you use the appropriate data structure to contain your data and the appropriate looping structure to display the data on your HTML pages. You will also use AJAX to populate all common components in your pages (site header, Navbars, site footer, etc.).

Instructions:

1. You will replace the paragraph data that is hard-coded in your current document with data that is read from a JSON file using AJAX. **(4 Marks: GUI, 12 Marks: Functionality):**
    a. Create a **JavaScript** function that uses AJAX to read a JSON file that contains all your paragraph data (4 Marks: Functionality).
    b. The data that is retrieved from the JSON file (**paragraphs.json**) should be stored in an **Array** Data Structure (4 Marks: Functionality).
    c. Populate the pages of your Mini-Portfolio site with this data. Ensure that you use the appropriate looping structure. (4 Marks: GUI, 4 Marks: Functionality).
2. You will inject HTML for your site header, Main Navigation and your site footer **(8 Marks: Functionality).**
    a. Create a new file, **header.html** and populate your page header which should include your main navigation (2 Marks: Functionality).
    b. Create a new file, **footer.html** and populate your footer bar which should include a copyright statement and any other appropriate information (2 Marks: Functionality)
    c. Create appropriate JavaScript methods and code that uses AJAX to load and inject your header.html and footer.html into your page. (4 Marks: Functionality).
3. Your site will be organized into the following structure **(8 Marks: Site Structure):**
    a. **Content** folder. This folder will contain your site's CSS files (0.5 Marks: Site Structure)
    b. **Scripts** folder. This folder will contain your site's JavaScript files (0.5 Marks: Site Structure)

c. You will include the Bootstrap CSS Framework in each of your HTML files (0.5 Marks: Site Structure)
d. You will include the Font-Awesome Icon Font Library in each of your HTML files (0.5 Marks: Site Structure).
e. Create a **paragraphs.json** file in your **Scripts** folder that contains your paragraph data in JSON format (2 Marks: Site Structure).
f. Include a **Web.config** file in the root of your site that defines the json **mimeType** property allowing Microsoft Azure to server static JSON files (1 Marks: Site Structure).
g. Include a folder named **Views** in the **root** of your site. Inside your **Views** folder create another folder called **partials**. You will populate the **partials** folder with your **header.html** and **footer.html** (1 Marks: Site Structure).

4. Include **Internal Documentation** for your site **(6 Marks: Internal Documentation):**
    a. Ensure you include a **comment header** for your **CSS and JavaScript files** that indicate: The **File name**, **Author's name**, **Student Number**, **web site name,** and **file description** (2 Marks: Internal Documentation).
    b. Ensure you include a **section header** for all of your **HTML structure, CSS style sections,** and any **JavaScript functions** (2 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 various 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**. The use of **GitHub pages** is recommended (4 Marks: Cloud Hosting).

SUBMITTING YOUR WORK

Your submission should include:

1. A zip archive of your website's Project files
2. A link to your GitHub repository.
3. A link to your live mini portfolio site hosted with a Cloud provider (or school server)

Feature	Description	Marks
GUI / Interface Design	Display elements meet requirements. Appropriate spacing, graphics, colour, and typography used.	4
Functionality	Site deliverables are me and site functions are met. No errors, including submission of user inputs.	20
Site Structure	Well organized site files. Separate HTML and CSS. Appropriate links to external documents and code. Code is error free. JavaScript libraries use a CDN.	8
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.	6
Version Control	GitHub commit history demonstrating regular updates.	4
Cloud Deployment	Deploy site to Cloud Service	4
Total		**46**

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.