WEBD6201 – Client-Side Scripting

Lab 4
Express Portfolio Site

Due: Week 12 (Saturday April 4, 2020) @ midnight

Value 12.5%

Express Portfolio Site **Maximum Mark: 53**

Overview: Working alone or with a partner, you will create a Fictitious Portfolio Website using **ExpressJS** and implementing the **EJS templating engine**. Your site must be hosted live on a cloud service such as **Microsoft Azure**, **Heroku,** or **Digital Ocean**. As an option, if working alone, you may use this assignment as a chance to craft your own personal portfolio website.

Instructions:

This Express site must include the pages from your Personal Portfolio 5 pages – your **Home page**, an **About Me** page, a **Projects page**, a **Services page**, and a **Contact Me** page.

1. You Site must include the appropriate content for a **Personal Portfolio (12 Marks: Content)**
    a. You must include a **Navigation Bar** or other Navigation scheme that allows the user to view each page of your site. (1 Marks: Content).
    b. You must include a **Custom Logo** for your site, this should be placed in or around the main Navigation bar. The **Custom Logo** can be as simple or artistic as you desire (e.g. you could use a primitive colour-filled shape like a triangle or hexagon with your initials positioned inside). Please do not use a logo that belongs to another company or person. (1 Marks: Content).
    c. Your **Home Page** should include some sort of welcome message and link or button that allows the user to redirect to the About Me Page and / or other pages. I recommend also including some sort of **Mission Statement** (2 Marks: Content).
    d. Your **About Me Page** should include the fictitious persons **name**, an their **image** (I recommend a head and shoulders shot that you find or create), and a short paragraph about who the person is. If you are using this assignment to create a personal portfolio site keep this clean and simple as it may be viewed by perspective employers. (1 Mark: Content)
    e. Your **Projects Page** should include images and information for at least 3 Projects you wish to highlight. These could be current projects you are working on or past projects you have

completed. If creating this for a fictitious person, link to three projects that already exist (you may find interesting projects to highlight on the internet). Include an image for each Project and a short description of your role (if any) and the outcome. (2 Marks: Content).

f. Your **Services Page** should include a short list of services you offer (e.g. general programming, web development, mobile apps, etc.). I recommend including images that make this more appealing to view. (2 Marks: Content).
g. Your **Contact Page** should include the person's contact information in a panel or other construct. Again, you may include your own information if this site is for your portfolio (1 Mark: Content).
h. Your **Contact Page** should include a short interactive form that allows the user to send the owner a message and provide basic contact information (First Name, Last Name, Contact Number, Email Address, Message, etc.). This form does not have to be fully functional initially. However, it should be able to capture the information entered by the user and redirect them back to the Home Page (2 Marks: Content).

2. Your Portfolio site will use **ExpressJS** and **NodeJS** and your site's content has been split across various **partial** files. You will integrate these partial files into the pages requested by the user by using the **EJS template engine**. You will use the existing index.js **View template.** You will create any required routes by using the Express middleware framework **(2 Marks: GUI, 8 Marks: Functionality):**
    a. Each page in the web site will be split into its own partial file and reside in the **Views/partials** folder (4 Marks: Functionality, 2 Marks: GUI).
    b. An Express Route must exist for each page of your site. **Note**: You will need to use the **router.get(path, callback(req, res, next))** method structure with a **res.render(view, locals)** method call to render each view (4 Marks: Functionality).
    c. Your **page logic** should reside in a controller file named **index.js** that will reside in the controllers folder (2 Marks: Functionality).
3. Your site will use the new structure created by the **Express Generator**. Your site files will be migrated to work within the **public**, **routes** and **views** folders **(3 Marks: Site Structure):**
    a. Generate your site structure with the Express Generator. **Note**: You must use the **-e** option to ensure that you implement the **EJS templating engine** for Express (2 Marks: Site Structure).
    b. Your **JavaScript**, **CSS and Multimedia Asset Files** should be moved to separate folders within the **public** folder. Using the Twitter Bootstrap CSS framework is strongly recommended, though not required. **Note:** the **public** folder is part of the path and does not have to be referenced (2 Marks: Site Structure).
    c. All Your Code (HTML, CSS, JavaScript, jQuery, etc.) is error free (2 Marks: Site Structure).
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 headers** 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).

SUBMITTING YOUR WORK

Your submission should include:

1. A zip archive of your website's Project files uploaded to DC Connect
2. A link to GitHub repository. Ensure that each partner is a collaborator.
3. A link to your live portfolio site hosted with a Cloud provider or your choice.

Evaluation Criteria

Feature	Description	Marks
Content	Appropriate Content is added to each page of your portfolio site	12
GUI / Interface Design	Display elements meet requirements. Appropriate spacing, graphics, colour, and typography used.	2
Functionality	Site deliverables are me and site functions are met. No errors, including submission of user inputs.	8
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. 4 marks for initial deployment to Cloud Host	6
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. 2 marks for initial commit	4
Cloud Deployment	Deploy site to Cloud Service. 2 marks for initial deployment	4
Total		**41**

This assignment is weighted **12.5%** 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.