# COMP125 – Client-Side Web Development

# Assignment 4
## Express Portfolio Site

Due: Week 14 (Wednesday August 15, 2018) @ midnight

Value 15%

Express Portfolio Site **Maximum Mark: 40**

**Overview**: Using your current portfolio site content from previous assignments, create your Personal 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**.

## Instructions:

This Express site must include the pages from your Personal Portfolio - 3 pages – your **Bio page**, a **Projects page**, and a **Contact** page.

1. Your Portfolio site will use **ExpressJS** and **NodeJS** and your web pages have been split to use different **View templates** and partials by implementing the **EJS templating engine** and Express Routes **(9 Marks: GUI, 4 Marks: Functionality):**
    a. Your site must include at least 3 View content pages: **bio.ejs**, **projects.ejs** and **contact.js.** Unlike previous assignments all page content should be embedded in the page (so no front-end JavaScript or AJAX required) (5 Marks: GUI).
    b. Your site should include at least 2 View templates – one for your **Bio Page** and one for your **Contacts** page. **Note**: You may include additional templates as needed to accommodate other site pages (4 Marks: Functionality, 4 Marks: GUI).
    c. An Express Route must exist for each page of your site. **Note**: You will need to use the **app.get(path, callback(req, res, next))** method structure with a **res.render(view, locals)** method call to render each view (4 Marks: Functionality).
2. 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 **(15 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 (3 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 (3 Marks: Site Structure).

c. You will define routes for all your site pages in the **index.js** file in your **routes** folder (1 Mark: Site Structure).
d. Your **view**s folder will contain your **EJS page templates**. You will create a separate folder named **partials** for all your partial **EJS files**. You will need several partials including **header.ejs**, **content.ejs**, **footer.ejs** and others. (6 Marks: Site Structure).
e. All Your Code (HTML, CSS, JavaScript, jQuery, etc.) is error free (1 Mark: Site Structure).
f. Your site uses appropriate **JavaScript** libraries that download via **npm** or function through the use of links to a various **CDNs** (Content Delivery Networks) (1 Mark: Site Structure).

3. Include **Internal Documentation** for your site **(4 Marks: Internal Documentation):**
    a. Ensure you include a **comment header** for your **HTML**, **CSS and JavaScript files** that indicate: the **File name**, **Student Name**, **Student Number**, **Date Created**, **web site name,** and **Latest Revision History** (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)
4. 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
2. A link to GitHub (preferable).
3. A link to your live portfolio site hosted with a Cloud provider

## Evaluation Criteria

| Feature | Description | Marks |
|---|---|---|
| GUI / Interface Design | Display elements meet requirements. Appropriate spacing, graphics, colour, and typography used. | 9 |
| Functionality | Site deliverables are me and site functions are met. No errors, including submission of user inputs. | 4 |
| 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 | 15 |
| 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. | 4 |
| 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** | | **40** |

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.