

Summative Assignment

Module code and title	COMP1111 Programming (Gold)
Academic year	2023-24
Coursework title	Assignment 1: Client-side scripting
Coursework credits	8 credits
% of module's final mark	40%
Lecturer	Ryan Crosby
Submission date*	Tuesday, December 05, 2023 14:00
Estimated hours of work	16 hours
Submission method	Gradescope (code)
Additional coursework files	NA
Required submission items and formats	<i>In a zip file : index.html, any additional HTML pages, CSS, Javascript files, JS out files, documentation and git log files.</i>

* This is the deadline for all submissions except where an approved extension is in place.

Late submissions received within 5 working days of the deadline will be capped at 40%.

Late submissions received later than 5 days after the deadline will receive a mark of 0.

It is your responsibility to check that your submission has uploaded successfully and obtain a submission receipt.

Your work must be done by yourself (or your group, if there is an assigned groupwork component) and comply with the university rules about plagiarism and collusion. Students suspected of plagiarism, either of published or unpublished sources, including the work of other students, or of collusion will be dealt with according to University guidelines (<https://www.dur.ac.uk/learningandteaching.handbook/6/2/4/>).

Summative Assignment 1:

Client-side scripting

Markers:

Ryan Crosby (ryan.crosby@durham.ac.uk)

Hand-out date: 26 October 2023

Components marked: code, documentation, git log.

Total marks: 100

Weight: 40% of module mark

Coursework submission

Deadline **05 December 2023**

Format	Gradescope: Source code (all compressed in a ZIP directory with correct file structure), including:
	<ul style="list-style-type: none">- README.txt with execution instructions- HTML, CSS and any media- client-side JavaScript- <i>.eslintrc</i>- Dataset used- JSDoc out html files
	All code, comments and website documentation must be submitted in English.
Plagiarism, collusion	Students suspected of plagiarism, either of published work or work from unpublished sources, including the work of other students, or of collusion will be dealt with according to Computer Science and University guidelines - https://www.dur.ac.uk/learningandteaching.handbook/6/2/4/
Chat GPT	For this assignment, I would strongly recommend not using generative AI for this coursework. The aim of this coursework is for you to learn and use the introductory concepts of programming which you will use throughout your degree. Completing the work yourself will develop the skills which will be used throughout your degree and career. Furthermore, using generative AI without citations and acknowledgement that it is not your own material will result in assessment irregularity procedures.

1. Learning Outcomes

Subject-specific Knowledge

- Interaction between JavaScript programs and the Document Object Model (DOM).
- Using control statements to loop and make decisions.
- Knowledge and understanding of good programming practice (e.g. reuse, documentation and style).

Subject-specific and Key Skills - students will be able to demonstrate:

- an ability to realise solutions to problems as working JavaScript programs.
- an ability to interpret project requirements and specifications.
- an ability to apply reuse by exploiting predefined components.
- an ability to use software tools related to programming (e.g., web frameworks, IDEs)
- an ability to communicate technical information.

2. Outline of requirements

In this coursework, you should demonstrate your **client-side JavaScript** skills by designing and developing a **website**/web application using one of the following domains

- [Books Data Set](#)
- [Board Games Data Set](#)
- Choose your own (if you choose your own check with ryan.crosby@durham.ac.uk To ensure domain and data set are appropriate.)

Section 3 provides details about the requirements. Here is an overview:

Using one of the identified data sets, the following must be completed:

- I. Create HTML & CSS
 - Ensure good usability design.
- II. Design web service and implement it in **JavaScript**. Demonstrate knowledge and skills in:
 - control statements, objects and functions
- III. Apply **good programming practices** by:
 - using tools that support software development, e.g. web frameworks.
 - writing documentation - by using [JSDoc](#).
 - accounting for code quality.

3. Instructions

Use Visual Studio Code and the live server extension.

Note: if visual studio code has not been downloaded on your computer you can find this via [Apps Anywhere](https://appsanywhere.durham.ac.uk/). URL: <https://appsanywhere.durham.ac.uk/>

Website functionality:

- Use the selected dataset to manipulate entities of your choice.
- **Input interface** - user should be able to input information (search, input text, select from options, etc.) that will update (e.g., filter) data visualisations.
- **Output interface** – changes that the user makes to the data should be reflected on the website. At least two output entities should be displayed on your website. This could be text, images or data visualisation. Justify your choices.

Website design and development skills:

- **Design:** Your website must have a clear **purpose**/aim which is explained/highlighted on the main/home web page.
 - o The website should be easy to navigate and use. E.g if you create more than one HTML file your website should include a navigation bar. All submissions must include an index.html file to act as your landing page.
 - o Apply user interface/experience **design guidelines**, you can refer to: <https://www.usability.gov/what-and-why/user-interface-design.html> and <https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-rules-of-thumb>. You should primarily focus on:
 - *usability* – ease of use, intuitiveness
 - *responsiveness* - adjusting for different devices (e.g., laptop and smartphone) and window size browsing.
- **Development:** Demonstrate skills in and use of:
 - o control statements
 - o objects and functions
 - o DOM
 - o embedding/importing data from JSON/CSV
 - o HTML+CSS (compliant with HTML5)
 - o website development tools - Bootstrap framework.

Code quality:

- Evaluate the quality of all your code by using **ESLint**.
- **Fix** the identified errors.
 - o Use the following to configure .eslintrc.js


```
module.exports = {
  "extends": "standard",
  "rules": {
    "semi": [2, "always"],
    "indent": "off"
  }
}
```

};

- Document your code with [JSDoc](#)
 - o Correct application: correct JSDoc comments are evident in JS code; html out files are submitted.
 - o Usability: Main methods and parameters were documented.
 - o References and acknowledgement of the source of original code and/or data (including licenses) were included. References should be included in code comments and should be in Harvard Form.
 - o Justification for choices made during the development of the website should be included either in the JSDoc comments or the git log messages.
- Use Git for version control and peer assessments. **Ensure that your repositories are private.** This is done by using git education.

4. Marking Criteria

The distribution of marks for Assignment 1 is presented in the table below.

	Mark /100
1. Website functionality	20
<ul style="list-style-type: none"> - Reading and storing data from a JSON / CSV file. (10p) - Manipulating / searching the stored data via inputs which interact via DOM (5p) - Outputs of appropriate forms that the user can interact with, should be included for the data manipulation. This can include text hovering, image generation, data visualisation. (5p) 	
2. Website design and development skills	30
<ul style="list-style-type: none"> - Purpose/aim of the website was highlighted and applied to relevant domains (3p) - <i>Usability</i> – ease of use, intuitiveness (2p) - <i>Responsiveness</i> - adjusts to different window size browsing (2p) - Appropriate use of UI/experience design principles (2p) - Development skills were demonstrated in: <ul style="list-style-type: none"> o control statements (4p) o objects and functions (5p) o DOM (4p) o embedding/importing data from JSON/CSV (3p) o website development tools, e.g. Bootstrap framework (5p) 	
3. Code quality	10
<ul style="list-style-type: none"> - There is evidence of applying the specified ESLint rules (2p) - Eslint test showed errors/quality issues were fixed (2p) - HTML+CSS (compliant with HTML5) (2p) - JSDoc out html files were submitted (1p) - Correct use of JSDoc comments in JS code (1p) - Main methods and parameters were documented (1p) - Clear referencing and acknowledgment of data/code sources (1p) 	
4. Git and Justifications	10
<ul style="list-style-type: none"> - Consistent and well-timed git logs throughout the time of the module. (Can include formative logs) (5p) 	

<ul style="list-style-type: none"> - Appropriate log titles and messages in your git log (included in first 5 marks). - Justification for inputs and output choices in either git log or program comments. (5p) 	
5. Beyond the specification	20
<p>This collection of marks will elevate your project into the 80%-100% range. This can include additional features such as: data visualization, accessibility features, classes with constructors, prototypal inheritance, code reuse, navigation bars, etc. Justifications for additional features and why they were appropriate for your data set should be included.</p>	
6. Participation	10
<p>10% of this assignment's mark will be derived from your on-going participation and engagement in the module's activities and tasks, including: formative assignment, peer review, quiz completion and discussion engagement. Counted as submitted (1) or not-submitted (0).</p> <ul style="list-style-type: none"> - Formative assignment (3p) - Peer review (3p) - Quiz Completion (2p) - Discussion Board Engagement (2p) 	

Testing environment. Your submission will be tested on:

- Windows 10 with Chrome Version 117
- Visual Studio Code Version 5.3

FAQs

1. How is this assignment linked with the formative assignment.
This assignment builds upon the formative work you did. You should have feedback from peers about your work (and given feedback to peers) as well as general feedback from your module leader. Ensure that you use this guidance going forward with your summative assessment.
2. Can I use node.js / npm for this development?
Yes, if you wish to use these as your 'additional features'. Please ensure that you include a readme file informing the marker how to run your features. Note the marker should not have to install anything on their machine to get your code to run.
3. Does the time difference between git commits matter?
Yes, we are expecting to see consistent engagement with the material. It does not have to be uploaded every day, however all development happening in the final 24 hours before submission will not gain these marks.