COMP30670 Assignment 4
Review Checklist
Name:
Student Number:
Date:

Code:

1. Does the code work? Does it perform its intended function, the logic is correct etc.

   There is no need to verify the other project's code runs on your own machine. Check the screenshots and confirm they match your understanding of the code. Check for the various elements necessary (html page with google maps loading, javascript to call an (flask) API, python code (flask), database queries, scraping code).

2. Is all the code easily understood?

   Comment on the readability of the code. How well can you understand the function? Can you easily see the program/data flows?

3. Does it conform to your agreed coding conventions? These will usually cover location of braces, variable and function names, line length, indentations, formatting, and comments.

   Comment on the overall design and structure of the code. Is it well formatted, are the variable names descriptive, are the functions well organised etc.

4. Is there any redundant, commented or duplicate code (remember DRY, YAGNI)?

5. Is the code modular?

   Comment on the different elements of the code (flask, python, database, html, javascript)- how well are the functions separated, could they be easily reused in other projects?

6. Control flow and programming.

   Can you spot any bugs in control flow? Loops not terminated correctly? Incorrect logic in control flow (if/while statements), string handling, error handling, bad variable names, convoluted flow, etc.

7. Can any of the code be replaced with library functions?

   Comment on any code which could be replaced- eg. some functions in python might be in a standard library, or pandas, or in Javascript there could be a better jQuery function.

8. Can any logging or debugging code be removed?
   Are there log statements (eg. 'print' in python, or 'console.log' in Javascript)?

Documentation:

1. Do comments exist and describe the intent of the code?

2. Comment on the project description? How well does it describe the overall function of the project and how well does it match your understanding from reading the code?

3. Are all functions commented?

4. Is any unusual behaviour or edge-case handling described?

Testing:

1. Is the code testable?
   Comment on how the design of the code affects its testability.

2. Do tests exist and are they comprehensive?

Miscellaneous:

1. Examples of elegant code/design?
2. Creative solutions?
3. Good Architecture/Design?