TEAM X-CEL - CODE REVIEW

11.12.2018

—

D H A R M I K SHAH
C A L E B CHEN
B Y R O N LEUNG
R A L P H MAAMARI

TABLE OF CONTENTS

INTRODUCTION

This document will highlight our code review strategy, and also will list each members findings based on the code that they are assigned.

CODE REVIEW STRATEGY

What to look for:

- Readability and Maintainability
    - Functions that are too large (should be separated)
    - No useless code (e.g. HTML tags that do not affect anything)
    - Comments explaining complex pieces of code
    - No unnecessary comments for things that are obvious
    - Consistent naming conventions (e.g. camel case)
    - Consistent code formatting (e.g. use of whitespace)
    - Descriptive variable names (it should be clear why each variable is used)
    - Redundant code that could be made into reusable functions

- Code Design
    - The code should follow MVC principles (the view is an .ejs file [html], which in turn should call a model .js file [jQuery], which communicates to the backend API controller, which is a .js [node] file
        - Model (JS)
        - View (.ejs)
        - Controller (JS)
    - Are there files/functions that are in the application, but never called?

- Functionality
    - Check for dead code in the database access where we return and have code underneath it that will never execute
    - Are all database err checked for, and have an appropriate response when it does occur
    - No leftover code used for debugging (e.g. console.log), or at least, no unnecessary debugging code left
    - Is the user alerted when they give an input the application cannot handle
    - Is the functionality of the code easily accessed by the people that need it
- Testing
    - Tests are easy to understand (e.g, they have a well named, clear description)
    - Tests are idempotent
    - Tests have a 100% coverage (e.g, database tests should cover if the item is in the database/if it isn't, what happens if we try to access when it is not there, etc)

CODE REVIEW SUMMARY

Byron (Dharmik's code)

- Many console.log statements used for debugging are left in the code, they should be either removed or commented out
- All of the jQuery code is in one big function, they should be separated into smaller functions to improve readability.
- Some comments only explain what has happened in the function, but does not explain what the following code actually does (e.g. comment says "we cannot be in the login page", but doesn't explain the implications of that).
- Some if statement conditions and code are put in one line, making it difficult to read
- Use of whitespace is inconsistent (e.g. random blank lines, if statements and code in one line, large JavaScript objects with multiple properties in one line)
- Variables are assigned and accessed without being declared (e.g. "radio" in create-acc.js)
- Some variables names are not descriptive (e.g. "db and db2" in index.js)
- Some comments explain obvious things and are not needed (e.g. explaining click events in upload-files.js)
- If there is console logging and alert code that are meant to be left in, they should use consistent capitalization and punctuation.
- Some alerts for the user shown during an error are not descriptive and does not provide the user with relevant or useful information.
- There are many unused variables (VSCode should display them in grey).
- There are event handlers that have no internal code and do nothing (e.g. in generate-report.js and upload-files.js).
- Some error handlers do nothing other than log something to the console, should probably alert the user instead.

Dharmik (Byron's code)

- Many console.log statements in the uploadFileController, and most of them are commented out
- Need to add comments for the process of parsing the csv file, as it is not directly evident just based on the code itself (the XLSX npm package usage)
- Be more specific then to name variables "data", as this is used in multiple locations and it is hard to distinguish
- Need to also add comments for the process of calculating the report totals, as you are the only one who did this part, and it is important for the team to understand
- In the uploadFileController and generateController, you have too much code under one function. Consider breaking this up into multiple functions

Caleb (Ralph's code)

- Not fully covered (i.e. upload function is not covered).
- Not idempotent
    - Executing the tests when the DB is empty can have a different outcome as when the DB is full.
    - The tests are inputting values into the DB that are not being cleared once the tests are complete.

Ralph (Caleb's code)

- Accessibility - Doesn't follow the WCAG 2.0 standards for web accessibility.
- Code duplicated - The navigation bar at the top of all the pages is repeated code instead of an injected header file.
- No use of semantic tags - Used divs instead of the semantic header tag for the navigation bar.
- Formatting inconsistency - There are white spaces and random indents throughout the code.
- Didn't close some tags - Some
 tags don't have
