3A W1 Code Review Checklist

CHECKLIST

Code Reviewers will follow the checklist for their code review process for 3A W1 project.

Generic checklist

1. No single letter names assigned to variables
2. camelCase used for naming objects and functions
3. PascalCase used for naming classes
4. Acronyms are all uppercased in identifiers
5. No long functions identified in the code
6. No long lines identified in the code/comments
7. There is no commented-out code
8. No redundant or duplicated code
9. For numeric values calculations are preserved while assigning the value
10. No literal values assigned to variables
11. There no hardcoded values
12. No deep nesting of loops and conditionals
13. `/* ... */` for multi-line comments
14. No long list of function arguments
15. Commit message contains summary line at the top
16. Credentials are not stored in the code
17. External URLs are accessed over `https`

Checklist for Javascript and Typescript programming

1. Constants are not declared inside loops
2. No `var` used to declare variables
3. `const` is used for constant values
4. No anonymous functions passed as arguments
5. There are no callback functions where a promise or async/await can be used
6. There are no loose equality operators
7. No `eval()` for generating dynamic code
8. No `for ... in` loop for iterating over arrays
9. No inline Javascript in HTML templates
10. Development dependencies are not present in `dependencies` section in `package.json`
11. There is dedicated commit for `package-lock.json`
12. No `*` in `package.json` to specify library version
13. Typescript: No non-primitive boxed objects used for declarations
14. Typescript: `any` is not used for declarations
15. Exception on filesystem or database calls are caught and handled
16. Named function expressions are used instead of function declarations
17. Default parameter syntax is used rather than mutating function arguments
18. No variable assignments chain like `let a = b = c`
19. Only arrow functions are used for function definitions
20. Only `Error objects' are passed to `throw`

Checklist for Front-end Programming

1. Only pipeable operators used for constructing chain of RxJS operators
2. No nested subscriptions to observables
3. All observables are unsubscribed
4. Observables are subscribed from templates, not from components
5. The modules which are not required immediately upon application start are lazy-loaded
6. Safe navigation operator (?) is used while accessing an object property in a template
7. There is no inline styling in HTML
8. Errors on HTTP requests (4 types) are caught and handled
9. In CSS !`important` is used ***only*** when the style has to take precedence
10. In CSS, IDs are not used as selectors
11. In CSS, no deep nesting of selectors to specify an element
12. CSS files are included at the top of the Index file
13. Javascript files are included at the end of the Index file

Checklist Back-end and Node

1. Inline dependencies should not be tagged `required`
2. Calls are not blocked
3. SQL queries are not generated by concatenating user supplied strings
4. Sensitive information is not logged
5. Module is `required` using explicit file/folder path and not using a variable
6. Errors on HTTP requests (4 types) are caught and handled when connecting to external services

ITEMS NOT COVERED

The code reviewers will not be covering following points while working on the code review.

1. Framework or language features are used instead of writing custom code
2. The code implements correctly the design specifications
3. Design patterns if used are correctly applied
4. Code is readable and intelligible other than the review points listed above
5. Unit tests are testing the business logic correctly
6. Single responsibility principle for classes and functions
7. Loose coupling between components
8. Suitability of comments in the code
9. Other potential implementations in terms of other best practices
10. Identifying bugs
11. Optimisation and tightness of code
12. Appropriateness of the information logged

* * * *