What to look out for when reviewing PRs?

Readability

- Is the code readable and easy to understand?
- Is any complex code supplemented with comments or documentation?
- Naming
    - Have variables and methods been given meaningful names?
    - Are there any misleading variable names or have acronyms been used that is assuming prior knowledge?

Architecture

- Has the file or folder structure been considered?
- Have components and methods been constructed in a way that they could be reused?
- Have they used Semantic HTML elements instead of a general-purpose tag (i.e. div)? Semantic HTML help to provide meaning and context to the content it wraps, helping accessibility, search engine optimization, and clearer code structure.

Error handling

- Are all scenarios being handled? Or have only the successful paths been considered?

Terminal warnings

- Have any terminal warnings been introduced? These can be found in the browser console.
- For example, when using the map function and you forget to give each child a unique key prop this will result in a console warning.

State management

- Do you need to store the variable in the global state, like Redux, or would it be more appropriate to store in the component state.
- Do you notice any prop drilling?

Functional Vs Class components

- Should the class component be refactored to a functional component and utilise React Hooks?

Avoid unnecessary component re-renders

- Could the *useMemo* be used to memoize values and recompute only if a dependency changes?

- Could the useCallback hook be user to memoize function references?
- Are all props required?

Styling

- Are we prioritizing the use of components from our shared library?
- Is it responsive?

What do you look out for when reviewing a React pull request?