

General:

Comments:

- Use `//` for descriptions above individual functions/properties, and broad sections in the code
- Use `/*...*/` for file descriptions at the top of the file

Variable/function naming:

- Use camelcase in JSON files

Line styling:

- End line with a semicolon `;` in javascript files.
- Maximum line length of 80

Indentation:

- Use TAB (set to 4 spaces)

Spacing:

- Classes should be separated by a blank line from their internal functions.
- Functions aren't separated from their code by a blank line
- Remove excess spaces

Front End:

- *Variable/function naming:*
 - Use camelCase
- All imports should be capitalized, the file path they are from don't have to be.

```
import React from 'react';
import styles from '../style.css';
import Styles from '../style.css';
import SignUpFormComponent from './SignUpFormComponent';
```

- All react js files should start with a file description. Line up first letter of each line

```
/* A full display of the artifact detailing all information provided
   to the backend */

import React from 'react';
```

Major files such as “Sign Up”, “Edit Items”, “Add Items”, etc... should be made a `React.Component` class instead of a function. In those classes big render components should be given their own file i.e:

```
import SignUpFormComponent from './SignUpFormComponent';

class SignUp extends React.Component {

  render() {
    return(
      <SignUpFormComponent />
    );
  }
}
```

- Multiline JSX code should be separated into their own line:

```
<input type="textbox" name="firstName" value={props.data.firstName}
onChange={props.handleChange}

<input
  type="textbox"
  name="firstName"
  value={props.data.firstName}
  onChange={props.handleChange}
/>
```

Back End:

- *Variable/function naming:*
 - Use underscores (and all lowercase)
- Imports of libraries are NOT capitalised:

```
const express = require('express');
```

- Imports of classes from local files are capitalised:

```
const Firebase = require('./firebase.js')
```