

Software System Engineering
Final Year Project

SHIKSHA The Planner

Front-end Code Review Checklist

-By CALTOR TEAM

[Earl Jedrick Macalino, Glaiza Regonas, Umangi Patel]

1. Code Formatting

The front-end is aligned properly, the starting and end point of a block can easily be identified. Proper naming conventions are also used and for almost all codes fits in a 14-inch laptop aside from one page where the placeholder for a textbox where, one would scroll horizontally to read.

```
return (
  <div>
    <Navbar />

    <h1>Course</h1>

    <button onClick={showForm}>Click here to enter grade information</button>

    {(form) ? (
      <div className="course-form">
        <form onSubmit={handleOnSubmit}>
          <label>Desired Grade: </label>
          <input
            type = "number"
            name ="desiredGrade6"
            value={values.desiredGrades6}
            placeholder="Enter your desired grade"
            onChange={handleOnChangeForm}/>

          <label>Is there is Lab?</label>
          <input className="checkbox1" type="checkbox" onChange={handleChecked} />
          <br />
          <button type="submit">Submit</button>
        </form>
      </div>
    ) : null}
```

2. Architecture

For the front-end each page contains components that are placed in folders that corresponds with it's respective pages. CSS however are mostly in one, aside from the navigation bar and one CSS for the overall look of the page.

3. Coding best practices

For the front-end, there is no hard coding, constants are used at all times. Comments are placed in some files to indicate why we are doing and what we are doing. We think it's not necessary to have the functions comments because they are pretty straight forward with the names given to them.

4. Non-Functional requirements

Due to us being new to React JS and only having a small amount of time to learn, we have not been able to properly reuse components, there are codes that are basically a

repeat of themselves, an example is the course and lab pages, they are all identical to each other. Examples like those would have been a good way to make use of React properly.

We believe the front-end has a good readability though, appropriate variable names for variables and function. This way it is easy to identify which function and variables name are for what. This is also easy to test.

Sources

Template used: <https://www.evoketechnologies.com/blog/code-review-checklist-perform-effective-code-reviews/>. (Accessed April 9, 2020)