

Challenger λ

Code Review

Table of Contents

Table of Contents	1
Code Review Strategy	2
Code Review Summary	3
Jin Zheng	3
Salim Mansour	3
Joseph Dong	4
Kevin Bato	4
Hayden Huang	5

Code Review Strategy

Our team conducted our code review during our weekly team meeting the week of November 13-18. We picked out snippets of code which we deemed worthy of review from each member and briefly discussed the code.

Some criteria which we used to determine whether or not code was worth of review was:

- Code when committed crashed the app in master branch
- Code when committed contained minor errors
- Code was not modular / was too inefficient
- Code was too difficult to understand without proper documentation
- Code followed poor/different conventions from existing code

Our team then analyzed the problematic code and offered suggestions to change the code to function better within our app. We also offered the individual advice so that the same issues do not occur in the future.

Code Review Summary

Jin Zheng

The team agreed that Jin's code is not very modular, there are many cases where the code is very similar to other snippets of code with very little change. Jin is advised to approach the code from a more modular perspective so that the team can reduce recycled code.

```
19 -router.get('/getMC', function(req, res, next) {
20     console.log('Viewing MC Questions');
21     res.locals.connection.connect();
22     res.locals.connection.query('SELECT * FROM Question WHERE qType="MC"', function (error, results, fields) {
23 @@ -26,7 +54,7 @@ router.get('/getMC', function(req, res, next) {
26         res.locals.connection.end();
27     });
28
29 -router.get('/getAll', function(req, res, next) {
30     console.log('Viewing all questions')
31     res.locals.connection.connect()
32     res.locals.connection.query('SELECT * FROM Question', function (error, results, fields) {
33 @@ -36,7 +64,7 @@ router.get('/getAll', function(req, res, next) {
36         res.locals.connection.end()
37     })
38 }
```

Salim Mansour

Salim is was advised to be more careful before committing untested code. One particular example showed that the backend request receiving methods he wrote were using the wrong request method.

```
60 +router.get('/attempt/insert', function(req, res, next) {
61 +    console.log('Creating a new Attempt:');
62 +    var sqladd = 'INSERT INTO Attempt (studentUsername, grade) VALUES ?';
63 +    var values = [
64 +        [req.body.studentUsername, req.body.grade],
65 +    ];
66 +    res.locals.connection.connect()
67 +    res.locals.connection.query(sqladd, [values], function (error, results, fields) {
68 +        if (error) throw error;
69 +        res.send(JSON.stringify(results));
70 +    });
71 +    res.locals.connection.end()
72 +});
73 +
74 +router.get('/attempt/update/:id', function(req, res, next) {
75 +    console.log('Updating Attempt with id:' + req.params.id)
76 +    var sqlupdate = 'UPDATE Attempt SET studentUsername = "' + req.body.studentUsername + '" , grade = "' + req.body.grade + '"
77 +    WHERE quizKey = "' + req.params.id;
78 +    res.locals.connection.connect()
79 +    res.locals.connection.query(sqlupdate, function (error, results, fields) {
80 +        if (error) throw error;
81 +        res.send(JSON.stringify(results));
82 +    });
83 +    res.locals.connection.end()
84 +});
```

Joseph Dong

In general, Joseph's code is very good however he tends to commit code with typos which may affect the usability of our app. The team advised him to be more careful before committing and to be more conscious of typos before committing code.

```
@@ -41,7 +41,7 @@ class MultipleAnswerDisplay extends React.Component {
  41     'option1: ' + this.state.option1,
  42     'option2: ' + this.state.option2,
  43     'option3: ' + this.state.option3,
  44 -    'answer: ' + this.state.answer
  45   })
  46   return (
  47     <div>

2 react_frontend/src/components/question/shortAnswerDisplay.js
@@ -29,7 +29,7 @@ class ShortAnswerDisplay extends React.Component {
  29     const values = List([
  30       'index: ' + this.state.index,
  31       'type: ' + this.state.type,
  32 -     'value: ' + this.state.value,
  33       'answer: ' + this.state.answer
  34     ])
  35     return (
```

Kevin Bato

For Kevin we discovered that the code he was writing in order to route our views could have been simplified. We advised Kevin to refactor the routing of our application's front and back end in order to simplify the code required.

```
class MainSwitcher extends React.Component {
  render() {
    return(
      <Switch>
        <Route
          exact path="/" />
        <Route
          exact path="/generateApp"
          component={GenerateApp} />
        <Route
          exact path="/viewApp"
          component={ListQuestions} />
        <Route
          exact path="/signUpApp"
          component={SignUpApp} />
        <Route
          exact path="/questionApi" />
        <Route
          exact path="/userApi" />
        <Route
          exact path="/attemptApi" />
        <Route
          exact path="/quizApi" />
      </Switch>
    )
  }
}
```

Hayden Huang

Hayden was responsible for the CSS styling of our app, however we discovered a problem with the styling where the buttons were styled however the buttons' functionality was being distorted by the styling. Ex. The button was styled larger than the actual button link.

home
generate
view(alpha)
sign up

short answer	
Value:	
Answer:	