Milestone 4

Spill

Spring 2018 SW Engineering CSC648/848

Section 01 Team 06

05/17/2018

Peter Mutch (peter2mutch@gmail.com)
Satjit Bola
Alaric Gonzales
Lorraine Goveas
Albert Fernandez Saucedo
Harpreet Singh

History Table:

1. Product Summary

Our website is called “Spill.” Spill is a site for regular people to discover and report environmental issues in their neighborhood.

Spill allows users to:

- Search for issues by zip code, city, or location name
- Report an issue near them

Spill is unique in that it is designed specifically to crowdsource location-based environmental reporting, as opposed to sites like the EPA’s.
Spill can be found at http://sfsuteam06spilldemo.dnsd.info/.

2. Usability Test Plan

Function to be tested: Search

Test Objective:

The objectives are to test how easy it is to use the search function. The search results page should display information relative to what the user searched.

Test Plan:

System Setup: Working Browser: Chrome or Safari Firefox
Starting Point: The home page of the website
Task: Search for a location to see what posts are in that area
Intended User: Any resident of that area
Completion Criteria: See posts for the area searched
URL to be tested: http://sfsuteam06spilldemo.dnsd.info/

Questionnaire:

1. Searching for posts was intuitive

①	②	③	④	⑤
Strongly Agree	Agree	Neither	Disagree	Strongly Disagree

2. Search results were displayed in a well organized manner

①	②	③	④	⑤
Strongly Agree	Agree	Neither	Disagree	Strongly Disagree

3. Search errors guided the user in the correct direction

①	②	③	④	⑤
Strongly Agree	Agree	Neither	Disagree	Strongly Disagree

3. QA Test Plan

Test Objectives:

The objectives are to test the accuracy of the search function. A correct search query should display the proper results. An incorrect search query should tell the user they could not find any matching results and display nearby, popular, or recent results instead. The security of the site will also be tested for sql injections.

Hardware and Software Setup:

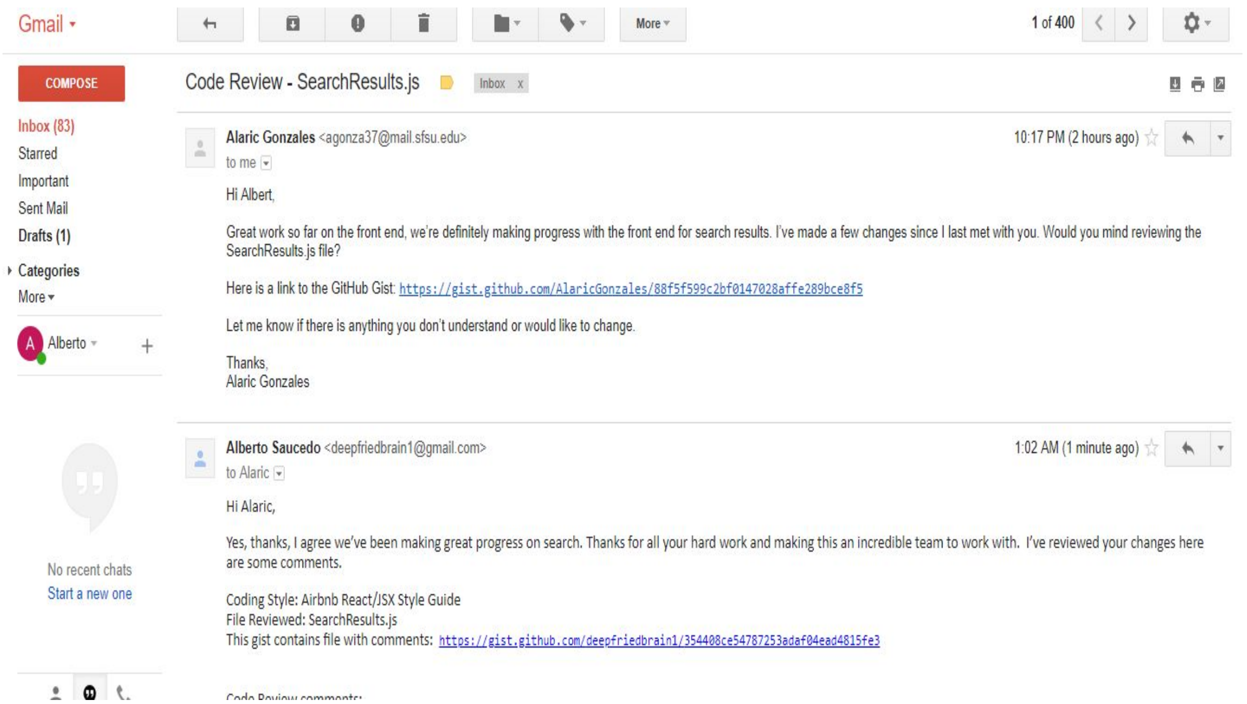
Hardware: A computer running Mac OS, Linux, or Windows
Software: Safari, Google Chrome, or Firefox

Features to be tested: Search

Test Cases:

Number	Title	Description	Input	Expected Output	Test Results(Pass/ Fail)
1	Correct Search Query	A correct search query which should display the appropriate results	Zion	1 search result Chemical Hazard at Zion National Park	
2	Incorrect Search Query	An incorrect search query should display a no results message and show similar results	Negative numbers, non-alphanumeric characters, etc	No results found, list of other reports	
3	SQL injection	A possible input from malicious users	Select * from hello;	Error message Other results should appear	

4. Code Review



Code Review - SearchResults.js

Alaric Gonzales <agonza37@mail.sfsu.edu>
to me
10:17 PM (2 hours ago)

Hi Albert,

Great work so far on the front end, we're definitely making progress with the front end for search results. I've made a few changes since I last met with you. Would you mind reviewing the SearchResults.js file?

Here is a link to the GitHub Gist: https://gist.github.com/AlaricGonzales/88f5f599c2bf0147028affe289bce8f5

Let me know if there is anything you don't understand or would like to change.

Thanks,
Alaric Gonzales

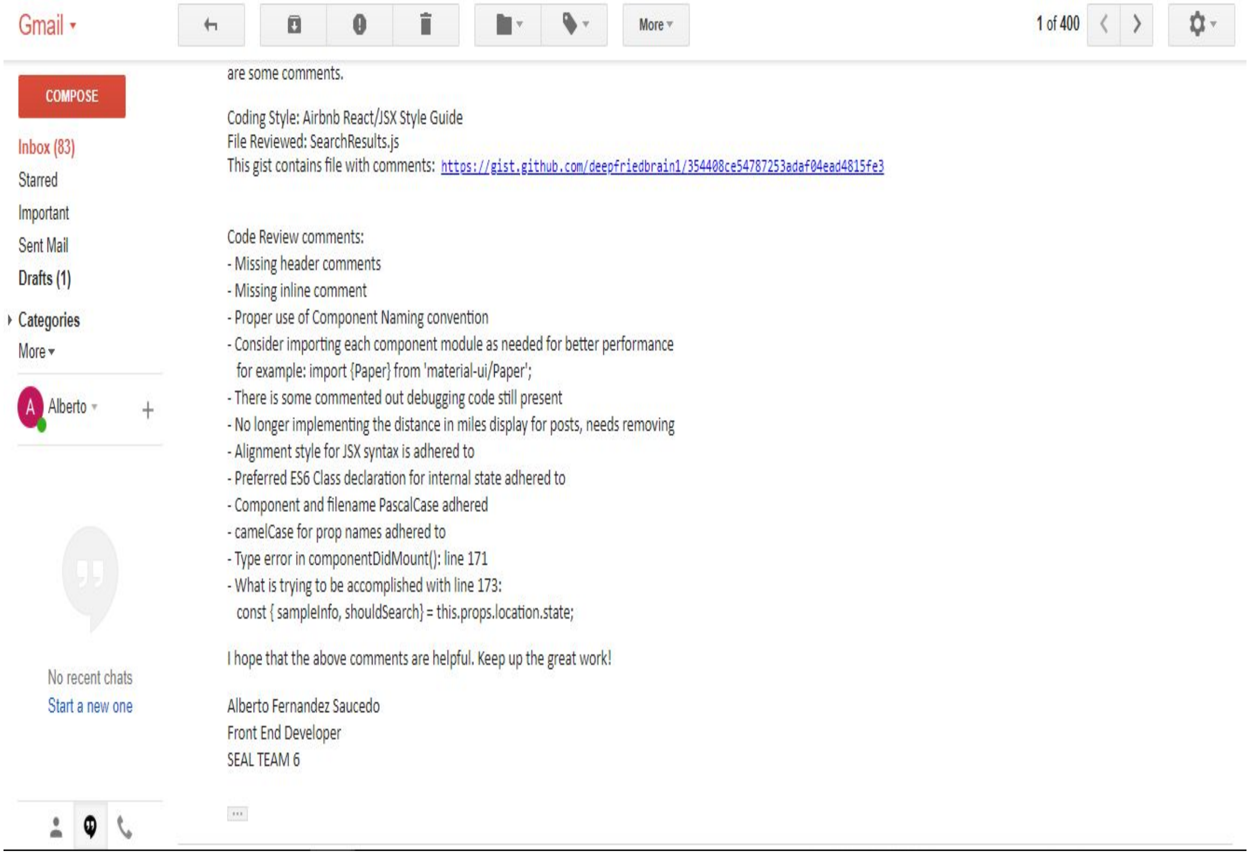
Alberto Saucedo <deepfriedbrain1@gmail.com>
to Alaric
1:02 AM (1 minute ago)

Hi Alaric,

Yes, thanks, I agree we've been making great progress on search. Thanks for all your hard work and making this an incredible team to work with. I've reviewed your changes here are some comments.

Coding Style: Airbnb React/JSX Style Guide
File Reviewed: SearchResults.js
This gist contains file with comments: https://gist.github.com/deepfriedbrain1/354408ce54787253adaf04ead4815fe3

Code Review comments:



are some comments.

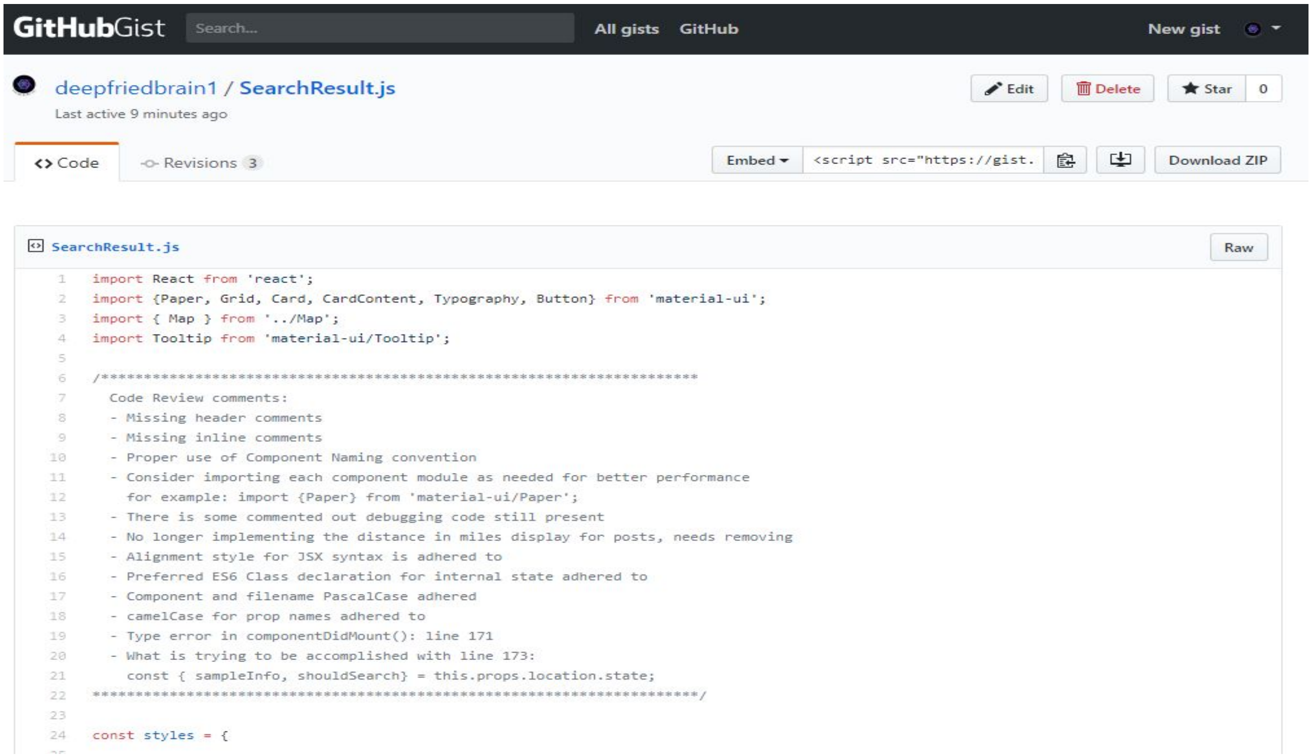
Coding Style: Airbnb React/JSX Style Guide
File Reviewed: SearchResults.js
This gist contains file with comments: https://gist.github.com/deepfriedbrain1/354408ce54787253adaf04ead4815fe3

Code Review comments:

- Missing header comments
- Missing inline comment
- Proper use of Component Naming convention
- Consider importing each component module as needed for better performance
  for example: import {Paper} from 'material-ui/Paper';
- There is some commented out debugging code still present
- No longer implementing the distance in miles display for posts, needs removing
- Alignment style for JSX syntax is adhered to
- Preferred ES6 Class declaration for internal state adhered to
- Component and filename PascalCase adhered
- camelCase for prop names adhered to
- Type error in componentDidMount(): line 171
- What is trying to be accomplished with line 173:
  const { sampleInfo, shouldSearch} = this.props.location.state;

I hope that the above comments are helpful. Keep up the great work!

Alberto Fernandez Saucedo
Front End Developer
SEAL TEAM 6

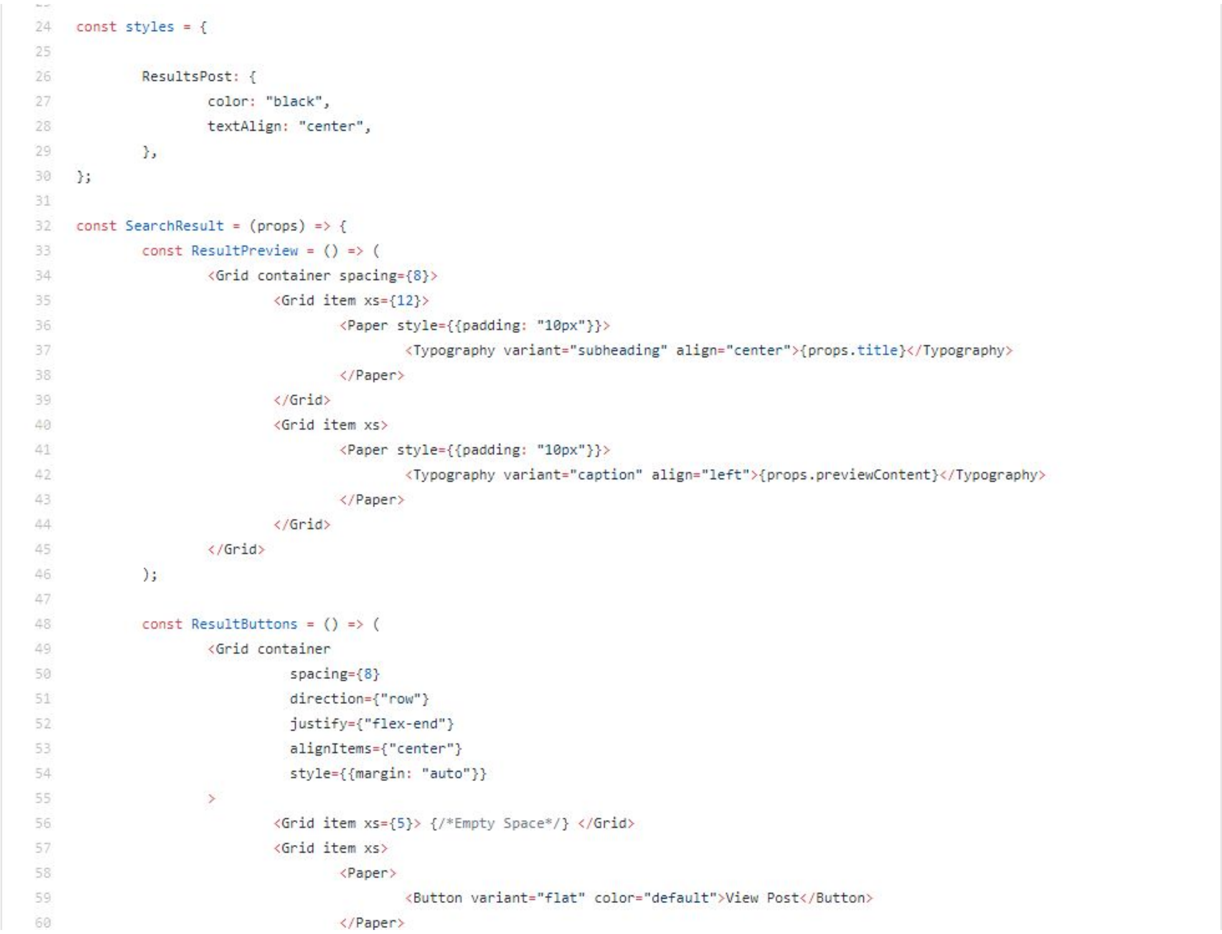


deepfriedbrain1 / SearchResult.js

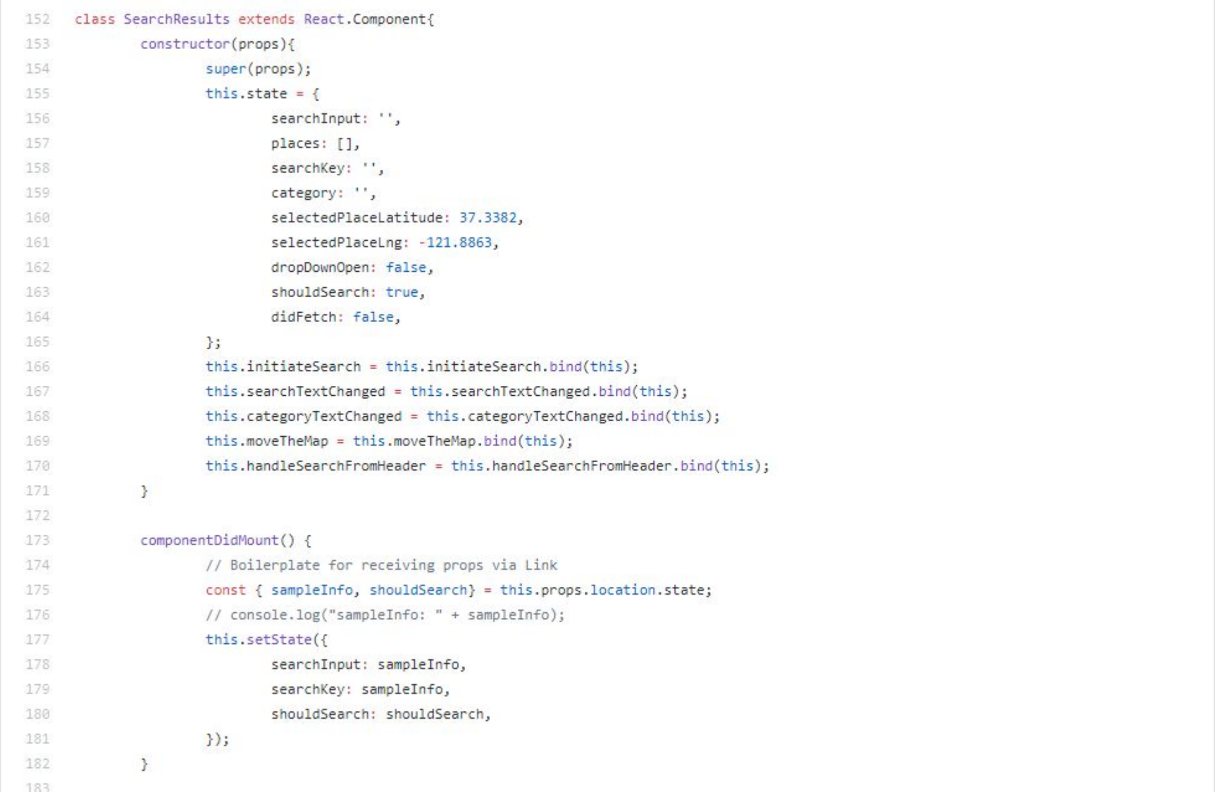
Last active 9 minutes ago

SearchResult.js

```
1   import React from 'react';
2   import {Paper, Grid, Card, CardContent, Typography, Button} from 'material-ui';
3   import { Map } from '../Map';
4   import Tooltip from 'material-ui/Tooltip';
5
6   /*******************************************************************
7     Code Review comments:
8     - Missing header comments
9     - Missing inline comments
10    - Proper use of Component Naming convention
11    - Consider importing each component module as needed for better performance
12      for example: import {Paper} from 'material-ui/Paper';
13    - There is some commented out debugging code still present
14    - No longer implementing the distance in miles display for posts, needs removing
15    - Alignment style for JSX syntax is adhered to
16    - Preferred ES6 Class declaration for internal state adhered to
17    - Component and filename PascalCase adhered
18    - camelCase for prop names adhered to
19    - Type error in componentDidMount(): line 171
20    - What is trying to be accomplished with line 173:
21      const { sampleInfo, shouldSearch} = this.props.location.state;
22  *******************************************************************/
23
24  const styles = {
```



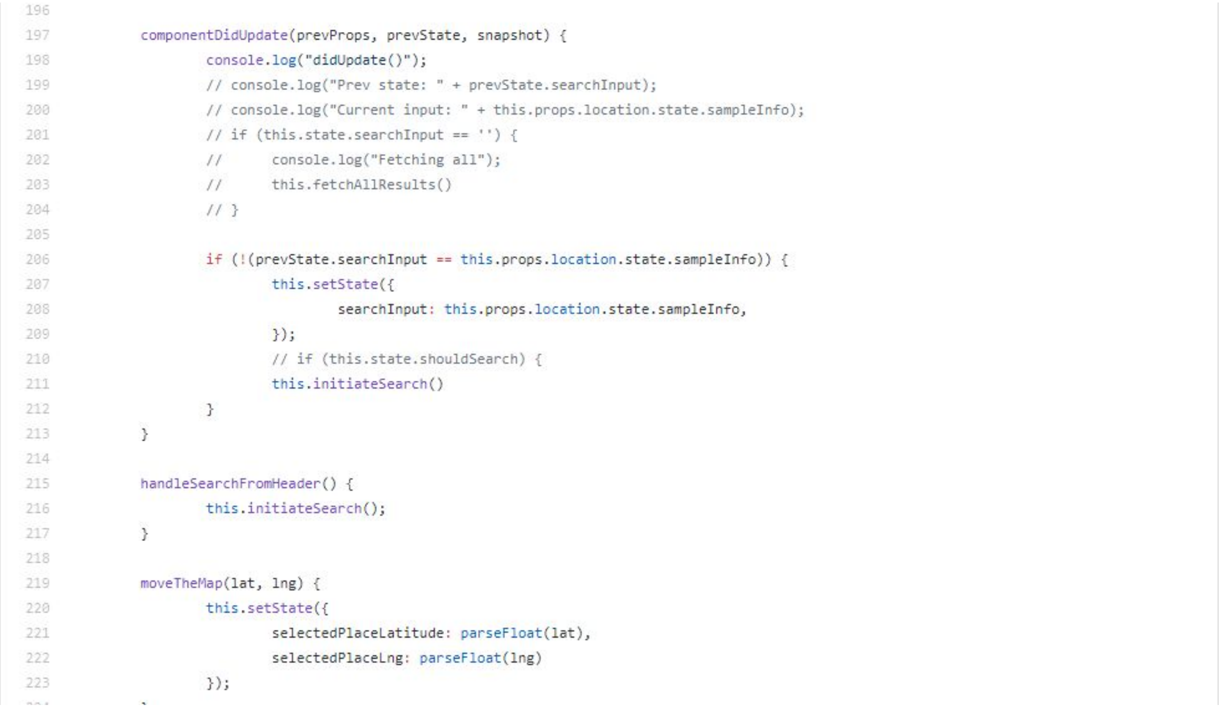
```
24  const styles = {
25
26          ResultsPost: {
27                  color: "black",
28                  textAlign: "center",
29          },
30  };
31
32  const SearchResult = (props) => {
33          const ResultPreview = () => (
34                  <Grid container spacing={8}>
35                          <Grid item xs={12}>
36                                  <Paper style={{padding: "10px"}}>
37                                          <Typography variant="subheading" align="center">{props.title}</Typography>
38                                  </Paper>
39                          </Grid>
40                          <Grid item xs>
41                                  <Paper style={{padding: "10px"}}>
42                                          <Typography variant="caption" align="left">{props.previewContent}</Typography>
43                                  </Paper>
44                          </Grid>
45                  </Grid>
46          );
47
48          const ResultButtons = () => (
49                  <Grid container
50                            spacing={8}
51                            direction={"row"}
52                            justify={"flex-end"}
53                            alignItems={"center"}
54                            style={{margin: "auto"}}
55                  >
56                          <Grid item xs={5}> {/*Empty Space*/} </Grid>
57                          <Grid item xs>
58                                  <Paper>
59                                          <Button variant="flat" color="default">View Post</Button>
60                                  </Paper>
```



```
class SearchResults extends React.Component{
        constructor(props){
                super(props);
                this.state = {
                        searchInput: '',
                        places: [],
                        searchKey: '',
                        category: '',
                        selectedPlaceLatitude: 37.3382,
                        selectedPlaceLng: -121.8863,
                        dropDownOpen: false,
                        shouldSearch: true,
                        didFetch: false,
                };
                this.initiateSearch = this.initiateSearch.bind(this);
                this.searchTextChanged = this.searchTextChanged.bind(this);
                this.categoryTextChanged = this.categoryTextChanged.bind(this);
                this.moveTheMap = this.moveTheMap.bind(this);
                this.handleSearchFromHeader = this.handleSearchFromHeader.bind(this);
        }

        componentDidMount() {
                // Boilerplate for receiving props via Link
                const { sampleInfo, shouldSearch} = this.props.location.state;
                // console.log("sampleInfo: " + sampleInfo);
                this.setState({
                        searchInput: sampleInfo,
                        searchKey: sampleInfo,
                        shouldSearch: shouldSearch,
                });
        }

```



```

        componentDidUpdate(prevProps, prevState, snapshot) {
                console.log("didUpdate()");
                // console.log("Prev state: " + prevState.searchInput);
                // console.log("Current input: " + this.props.location.state.sampleInfo);
                // if (this.state.searchInput == '') {
                //      console.log("Fetching all");
                //      this.fetchAllResults()
                // }

                if (!(prevState.searchInput == this.props.location.state.sampleInfo)) {
                        this.setState({
                                searchInput: this.props.location.state.sampleInfo,
                        });
                        // if (this.state.shouldSearch) {
                        this.initiateSearch()
                }
        }

        handleSearchFromHeader() {
                this.initiateSearch();
        }

        moveTheMap(lat, lng) {
                this.setState({
                        selectedPlaceLatitude: parseFloat(lat),
                        selectedPlaceLng: parseFloat(lng)
                });
        }
```

5. Self-check on best practices for security

The major assets that we are protecting include:
Registered user information: first name , last name, email , username and password
Posts: user id, address, location name, image

We are using Bcrypt to encrypt passwords in the database.
We are not validating any data input yet.

6. Self-check for adherence to original non-functional spec

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO). -**DONE**
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome. -**DONE**
3. Application shall have responsive UI code so it can be adequately rendered on mobile devices but no mobile native app is to be developed-**DONE**
4. Data shall be stored in the team's chosen database technology on the team's deployment server. -**DONE**
5. Application shall be media rich (at minimum contain images and maps)-**DONE**
6. No more than 50 concurrent users shall be accessing the application at any time -**ON TRACK**
7. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. -**DONE**
8. The language used shall be English. -**DONE**
9. Application shall be very easy to use and intuitive. -**DONE**
10. Google analytics shall be added -**ON TRACK**
11. No e-mail clients shall be allowed -**DONE**
12. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated. -**DONE**
13. Site security: basic best practices shall be applied (as covered in the class) -**ON TRACK**
14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development -**DONE**
15. The website shall prominently display the following exact text on all pages *"SFSU Software Engineering Project, Spring 2018. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application). -**ON TRACK**