Find and Share Trails

Ching-Wei Lin, Dajun Gu
GitHub Link: https://github.com/xingchengyusi/find-share-trails
Website Link: https://findsharetrails.netlify.com/#/

1. Decided to separate the project into two parts, the map and the database.
2. Looking for database, didn't want to use the normal database like MongoDB, so decided to use google sheet which is common and free. The best part is that its user authentication is already built in.
3. General local javascript has low performance when interacts with Google sheet, so decided to use google app script, which works with Google sheet perfectly.
4. Google app script works just like javascript but the communication between Google script and html file is different. It took some effort to figure out the usage from Google document.
5. Had some trouble connecting to these two parts. Finally, using redirect method to render each other.
6. Had some trouble on searching from month. Solved by mapping all the month into integer and simply used comparison logic.
7. For the react app router, I use "react-dom-router" because the switch function is really easy to use.
8. When I implement removing the markers in the google map section, I find that my marker cannot update with props changing. To update those markers, the original markers should be stored and then we can remove it when next time call the function "shouldComponentUpdate". In this case, I set the map object and marker objects to the state by "setState". However, new problems appear. Maximum update depth exceeded that mean "render" enter an endless loop because I set state in the "componentWillUpdate" or "componentDidUpdate". After the function "shouldComponentUpdate" run and return true, those three parts, such as "componentWillUpdate", "render", and "componentDisUpdate" will run. If the state is changed in those three parts, the React finds the component should update again when this update did not finish. In this case, the endless loop appears.
9. This problem makes me rethink my component structure. The component "Trails" contain two small components that are "Trails map" and "Trails list". The trails map is a google map and container several markers display the trailhead. The trails list is multiple pages, and display five trails one time. The function I should finish is when the trails list page changed, google map

should remove old markers and add new markers. Actually, the essential is changing “state” when “props” are changed.

10. One of the resolve methods is using “componentWillReceiveProps” function with the next props as a parameter. However, this is dangerous. It causes useless data, and the structure will be more complex. Also, from the React web site, it causes multiple data sources in some cases. Based on my condition, I chose to use a condition statement to identify whether the states should be updated.
11. For React, asynchronous is really a problem. After using the “setState” function, only the previous state can be used because react not really set it after this function end. It will combine some set functions and do them at the same time.
12. For another asynchronous problem is fetch data from API. The “await” is useful to wait for the progress done. Also, the function should be an “async” function if the “await” exists.