Week 3 Assignment

Your goal this week is to make a React application that makes use of the concepts of react that we are learning so far. (Components, Props, State, Modifying State (from child components), and React Router. This will not be easy, but feel free to reach out to me any time throughout the week for help. If you complete the assignment before Saturday, then you can present your work to the class. If you didn't complete it, don't worry, we can all work on it together :) Just do your best!

For the assignment, you will be recreating the application in the video I posted in the slack channel. Here are some general steps for accomplishing this:

- Create a new React Application using Create React App

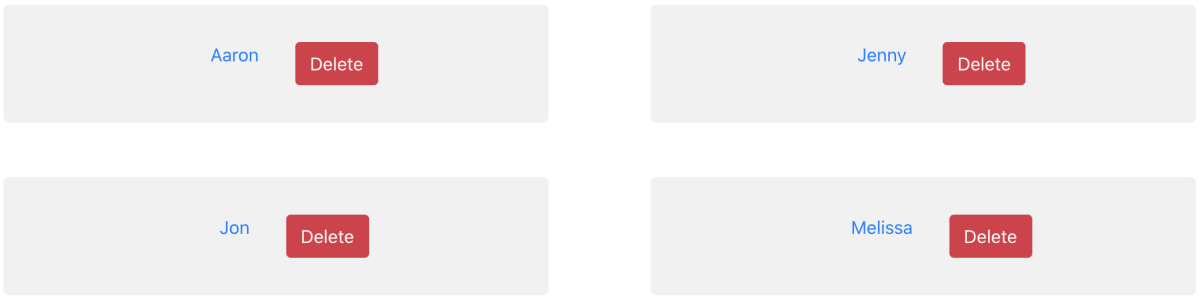
Initialize State

- Initialize some application level state to a parent component (this will be an array of objects that will look something like the following. Feel free to fill this array of objects with whatever you like). **Do not do this in the App.js file as App.js is where we will put our React Router implementation later.**

```
const initialState = [
  {
    id: 1,
    name: "Aaron",
  },
  {
    id: 2,
    name: "Jenny",
  },
  {
    id: 3,
    name: "Jon",
  },
  {
    id: 4,
    name: "Melissa",
  },
];
```

Pass state to a child component, map through, display

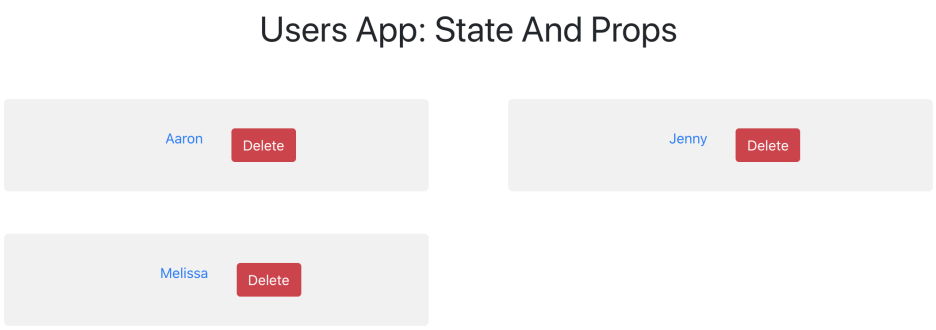
- Create another component that will take this state array as a prop. Inside of this component you will use the .map function on the array prop. For each of the iterations of the .map function, you will return some JSX that will display a value from your object. In my example I am displaying the **name** property of the objects. You will also need to give this JSX a delete button.



- Import this component into your parent component and display it using it's custom JSX tag. At this point, if you render your parent component, you should have a page with some JSX items that have been mapped out from an array of objects property.

Create a function to delete items from state, pass to child components, use as an onClick event

- Next you will go to your parent component that has the application level state (your initial state), and create a function that will modify your state. The function should take in an ID as a parameter and use the .filter function on the state array. As the .filter function iterates through your state array, you will write some code to filter out the object whose ID matches the ID you gave in the parameter.
    - If you are using a class based component, you will modify your state using this **setState** method, if you are using a functional component, you will do this by using the **useState** hook.
- Pass this function as a prop to your child component (the one that is mapping through and creating JSX elements).
- Now inside of your child component, you will add an onClick event to your delete button, which uses the function that you just passed in. If this was coded correctly, you can now temporarily delete items from your parent's state array by clicking on the delete button in a child component.



Now that we have the basics of our app working, we implement two different features of React Router.

React Router

- Begin by importing the following React Router requirements into your App.js File.

```
import {
  BrowserRouter as Router,
  Switch,
  Route
} from "react-router-dom";
```

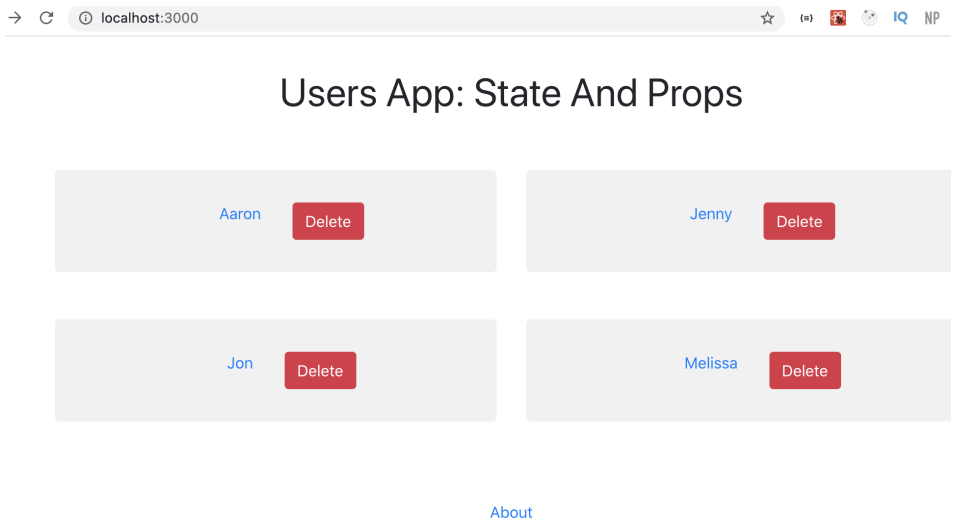
- Implement React Router by using the React Router Components. Your top level component should be the <Router> </Router> followed by the <Switch> </Switch> then the <Route />

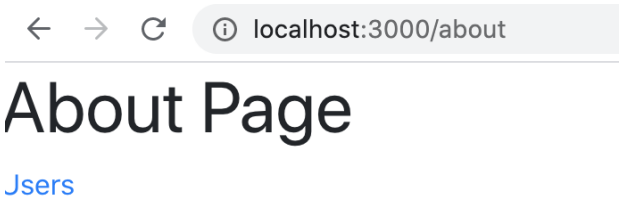
- Add a route that will render your Parent component that holds the application level state when you get to the "/" path. Make this an exact path.

- Next add a route that will render an About component when you reach the "/about" path. This About component can simply return an h1 tag that says "About". This is just to show you how React Router will let you navigate between pages.

    - Now if you go to your URL and type /about to the end of it, you should be able to see this component.

- Next, navigate to your Parent Component that is mapping through the state, and import the Link component from react-router-dom. Give this Link component a prop that links to the "/about" path. Now when you click on this link, you should be taken to the /about path, where the About component will be rendered by react router.





- Create a Link component on your About component that will take you back to the “/” route.

Bonus Exercise

- Create a dynamic route that will take you to a page that displays the object’s ID number. The path for this will look something like “/objects/:id”.
- To do this, you will wrap one of your mapped JSX items in a Link component and pass to the link path the objects ID.
- Hint: Research the react router “match.params” prop.