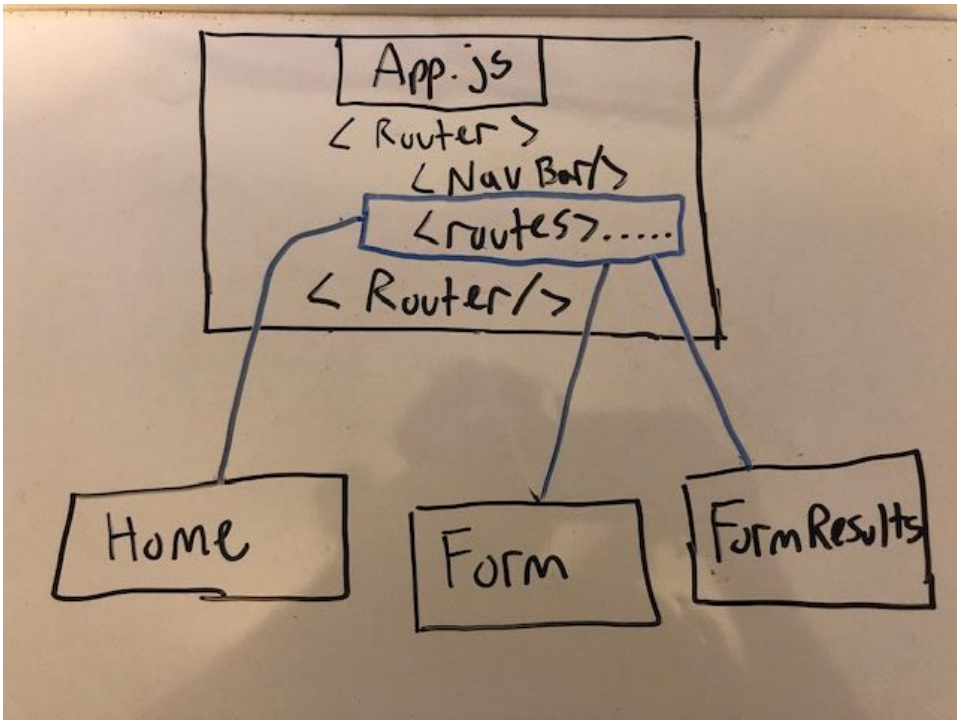
Task 1
Getting familiar with React Router and Components

Background:
We will most likely use React Router in our project. An easy way to think of React Router is it allows you to switch out React components based on URLs. React is hierarchical with how components are rendered to the screen. It can be seen as tree, especially in the case of this assignment's code.



App.js is the highest on the hierarchy, then Router, then NavBar, then routes etc.

This is the order of how this entire program will be rendered.
1.
App.js will be rendered first. This code can be found in index.js.

```
ReactDOM.render(<App />, document.getElementById('root'));
```

2.
Router is a built-in React component that doesn't render anything to screen.

3.
NavBar is a component with two icons that are links, the links go to '/' and '/form'. Since NavBar is rendered in App.js, it will appear once App.js is rendered and stay until App.js is gone. Meaning it will stay until you close the application.

4.
Route(s) are where we can dynamically change components viewed on the screen and are children of the Router component (Navbar is also a child of Router and a parent to Routes, which is why it will show up on every screen regardless of the Route). Depending on the URL specified by path="/somepath" it will render the component specified by component="somecomponent". You can think of it as that the component will replace that Route element. If you "npm start" the application and go to "localhost:3000" (homepage) it will show the {HomeText} component. Likewise, if you go to "localhost:3000/form" it will show the {Form} component.

```
class App extends React.Component {
  render() {
    return (
      <Router>
        <NavBar />
        <Switch>
          <Route exact path='/' component={HomeText} />
          <Route path='/form' component={Form} />
          {/* <Route
            path='/results'
            render={props => <FormResults items={props} {...props} />}
          ></Route> */}
        </Switch>
      </Router>
    );
  }
}
```

Task 1: Create your own component and Route.

a. Create your own file with a .js extension and create a javascript function that returns some basic HTML. You will need to export the function and import it into App.js. Google how to do this its not that hard. There is also an example in FormResult.js.
b. Make a new <Route> tag in App.js that renders your component. Name the path whatever you want. Save it and try to go to that path in the browser URL to make sure it loads.
c. Create a new icon link in the NavBar. Go to https://material.io/resources/icons/?style=baseline and pick an icon to use. The import names for material ui icons are usually very similar to how they are in NavBar. For

example on the site the home icon is called “home” but the import is `@material-ui/icons/Home`

Leave the route path=”/results’ commented out for now, that will be part of task 2.