**Notes on React**

Notes from YouTube course: https://www.youtube.com/watch?v=w7ejDZ8SWv8&t=464s

**PART 1 – Introduction:**

- React is a Javascript frontend framework that runs in the browser.  
- React builds a Single Page Application (SPA) where all the routing to the server is handled by React.  
- As React is a frontend framework, it cannot work directly with the backend database.  
- For React, look at the frontend as a bunch of components.  
- React components can be created with either classes or functions. For class-based components, must use extends React.Component and render() method.  
- Both class-based and function-based components return JSX (JavaScript Syntax Extension). JSX looks like HTML and accepts JavaScript expressions to make code dynamic.  
- State of a Component = An object that controls how a React component is rendered and behaves. Any data used by a React component is part of the component’s state.  
- Data that is shared across the whole React UI is either “App” or “global” state.  
- Before version 16.8, only class-cased components used React states. However, in later version, function-based components to use state.

**PART 2 – Creating a React app:**  
- One way to create a React app called “task-tracker” is by running the following command on the terminal:  
npx create-react-app task-tracker  
**This sets up all the packages and folders we need for a react application.**  
(The difference between npm and npx is npm would install the create-react-app command globally on your system. However, npx just runs the create-react-app command.)

**PART 3 – Files and Folders:**  
- Once the React application has been created, one of the files is package.json. One of the dependencies in package.json is “react-dom which is responsible for rendering the React app onto the DOM (Document Object Model) on the browser.  
(For React Native, it uses “react-native” instead of “react-dom”).  
- The dependency “react-scripts” provide the development servers.  
- To start the Dev server on VS Code, run the following on the terminal:  
npm start  
  
  
(If you run npm build, it builds a directory with a production build.)  
- In the “public” folder, there is the index.html which is the of SPA where all the UI is rendered. In the body of this index.html is div tag with ID = ”root”. This ID “root” references the element “root” in index.js in the “src” folder. This element is used by the method “reactDOM.react” and inserting the app into the “root” div. This app comes from the “import App from ‘./App;’ which contains the root app component. All other components will be placed into this root app component.

**PART 4 – App Component and JSX:**  
- In the “src” folder, the app.js folder is the main component in React and all other components will be contained inside this root component. The app component is a function-based component which returns JSX.   
- JSX (JavaScript XML allows you to write HTML in JavaScript).  
- JSX expressions must have only one parent element. So, every React Component can only return one parent element like a div. All other elements for this given component must be inside the parent element.

**PART 5 – Expressions in JSX:**  
- In a React component, you must use curly brackets like {} to embed a JSX expression. For example, to render the expression “name” in the function component “App”:

function App() {  
 const name = “Patrick”  
   
 return (  
 <div className=”container”>  
 <h2>Hello {name}</h2>  
 </div>  
 )  
}  
  
  
**PART 6 – Creating a component:**