**Understanding SRC folder:** This is the folder where we will spend majority of our time and in this folder, we will write React code. First thing you'll notice in the SRC folder is that we have two JavaScript files and one CSS file. We are just going to write some JavaScript code.

**Understanding index.js file:** This is the first code file, which will be executed whenever the page is loaded. So, whenever you visit **http://localhost:3000**, this index.js file is executed initially and it's one of the few things you need to remember. **NPM** start the development server and also watches our code, this process will not just watch our code and but deliver it to the browser, but before it delivers it will transform the code as well. So that certain features which are not supported by the browser will work in the browser. For example, the import below:

**import './index.css',** where we import a CSS file into a JavaScript file. That's not something which would work in regular JavaScript, that would be an invalid syntax. You can't import CSS into JavaScript. But here in this project set up, it does work. It tells this NPM start process, that we simply want to include this index.css file into our overall application. Another example, which is not a regular JavaScript syntax, would be **<App />.**

It looks like some kind of HTML code inside of a JavaScript file and normally this also wouldn't work. Again, here it does work but only because this is transformed before it's delivered to the browser.

**What's happening in index.js file?**

We are importing **ReactDOM** from **'react-dom'** which means we're importing something from the third-party library, which is one of our dependencies and it is downloaded and installed locally. In the **package.json** file, you actually see two React dependencies, react and react-dom. Technically these are two separate packages, you can think of them as React the library. It's split across two packages with different responsibilities but in the end, react-dom and react these two dependencies together form the React library.

**import ReactDOM from 'react-dom'**

In index.js, react-dom is exporting some ReactDOM object and we're importing it in the render() method of the index.js. The render() method takes two arguments, the second argument is a default JavaScript DOM API which we're calling on the global document object. We have a getElementById() method to select a certain DOM HTML element by its ID.

Now you might wonder which element? We have no HTML code. Well, there also is a public folder, this folder is a folder in which we will rarely work, but it holds one important file (index.html file). This is the single HTML file, which in the end is loaded by the browser. Because with React we build so-called single page application.

It bascially means that only one HTML file is delivered to the browser and hosted by the browser therefore rendered by the browser. So, we have one HTML file but what we see on the screen constantly changes because of the React. In index.html file we have **<div id="root"></div>** it's the only HTML element in the entire body except for the noscript element and this div has the id="root". It is this div which is selected by the code snippet document.getElementById('root')

And we are telling ReactDOM that we want to render <App /> inside of div with Id='root'.

**What's <App />?**

We're importing App from the App file and it's a JS file and you may omit and you should omit .JS extension in your imports. If it's another file, like a CSS file you have to add it but if it's a third-party library or one of your JS files, you can omit the extension. <App /> is JSX and it is a component. And if you take a look in the App.js file it has a function App() and we are exporting that function as a default from the file and then in the index.js file we are importing it.