React – Front-End and Directory Structure
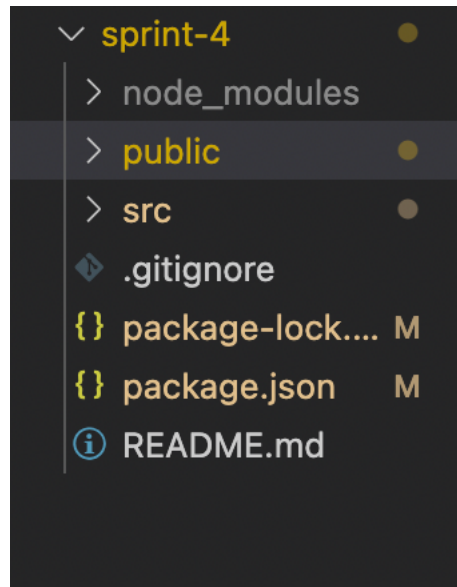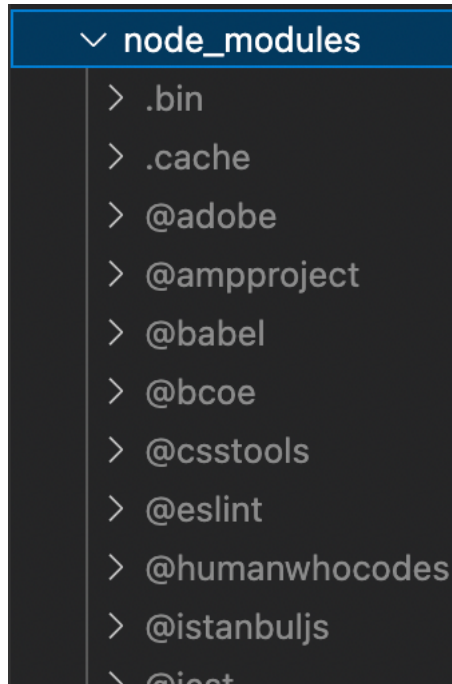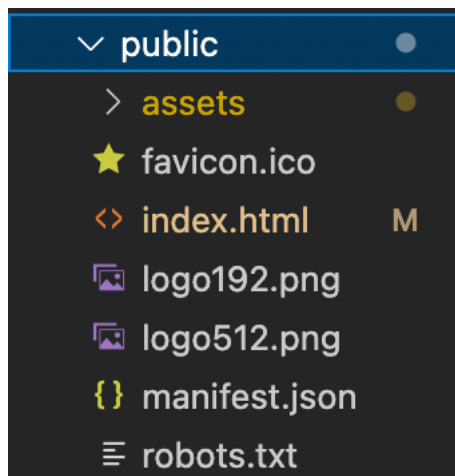
Directory Structure
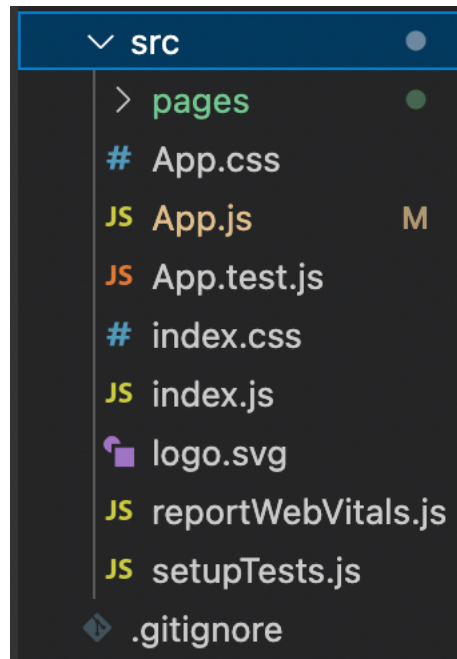
Full Directory Structure:

1. *gitignore file*: is a text file that tells Git which files or folders to ignore in a project.

2. *package.json*: is a **JSON file** that holds metadata relevant to the project and it is used for managing the project's dependencies, scripts, version. This file is basically used to give information to npm that allows it to identify the project as well as handle the project's dependencies.

3. *package-lock.json:* is automatically generated for any operations where npm modifies either the node_modules tree or **package**.json. It helps while a team is working on private apps to ensure that they are working on the same version of dependencies and sub-dependencies. It also maintains a history of changes done in package.json so, that at any point of time, when required previous changes can be looked back in the package-lock.json file.

4. *node_modules:* contains all dependencies and sub-dependencies specified in package.json used by React app.

5. *public:* This folder contains files that don't require any additional processing by webpack. The index.html file is considered as an entry point for the web application. In the HTML file you can import libraries such as CSS libraries and add meta tags.
    a. *Note:* Assets has been added here for the front-end template, it contains the images, css files, or js files necessary for the sites' styling
    b. *Index.*html: contains the header for the 'html' page

6. *src:* This folder is the main folder of React application as it contains JavaScript which needs to be processed by webpack. In this folder, you make your components, and then you can import them in app.js. Lastly, src contains registerServiceWorker.js which takes care of caching and updating files for the end-user. It helps in offline capability and faster page loading after the first visit.
   a. *Note:* pages has been added for the front-end template, it contains the 'html' for the pages that are on the site.
   b. *Index.*js: Contains the code that is ran for the front-end of the site (call pages in here)

https://soniapahuja75.medium.com/react-folder-structure-e05e23420a4f

<u>Front-End – Converting HTML to React JS</u>

public:
Step 1: create a directory called "assets" and copy and paste the css, js, images, directory/files into this folder so the site can access them.

Step 2: Copy and paste the header of the html file into index.html where the default header exists, and change path names to include the assets folder

src:
Step 3: create directory called pages and create js files for each page following this format:

```
import React, { Fragment } from 'react';

function Scheduler(){
    return(
        <Fragment>
            <body>
            <div class="container">
                <div class="table-responsive">
                    <table class="table table-bordered text-center">
                        <thead>
                            <tr class="bg-light-gray">
                                <th class="text-uppercase">Time
                                </th>
                                <th class="text-uppercase">Monday</th>
```

```
9                              <div class="font-size13 text-light-gray">Marta Healy</div>
0                          </td>
1                      </tr>
2                  </tbody>
3              </table>
4          </div>
5          </div>
6          </body>
7      </Fragment>
8      )
9  }
0  export default Scheduler;
```

The body will be the html necessary for the page.

Note: "class=" needs to be changed to "className=", all tags need to be closed (ex. ensure image tags), update the image tags to use the asset folder in the path

Step 4: Add the pages to the App.js file in this format

```
import './App.css';
// import Scheduler from './pages/scheduler';
import Welcome from './pages/welcome';
function App() {
  return (
    <div className="App">
      <Welcome/>
    </div>
  );
}
export default App;
```

https://www.youtube.com/watch?v=NSyCRfUzMb0