

Dependencies:

- "@emotion/react": "^11.9.0",
- "@emotion/styled": "^11.8.1",
- "@material-ui/core": "^4.12.4",
- "@mui/icons-material": "^5.6.0",
- "@mui/material": "^5.6.0",
- "@mui/styled-engine": "^5.6.1",
- "@mui/system": "^5.6.0",
- "@testing-library/jest-dom": "^5.14.1",
- "@testing-library/react": "^11.2.7",
- "@testing-library/user-event": "^12.8.3",
- "antd": "^4.16.13",
- "d3-format": "^3.0.1",
- "nodemon": "^2.0.15",
- "query-string": "^7.0.1",
- "react": "^17.0.2",
- "react-bootstrap": "^2.2.3",
- "react-dom": "^17.0.2",
- "react-router-dom": "^5.3.0",
- "react-scripts": "^4.0.3",
- "react-vis": "^1.11.7",
- "react-wordcloud": "^1.2.7",
- "shards-react": "^1.0.3",
- "styled-components": "^5.3.5",
- "web-vitals": "^1.1.2

Build Instructions

1. Before building, please ensure that the Application Structure contains the following folders and files. See the accompanying descriptions for context.

- a. **Server Folder (/server):**

- i. **.gitignore:** A gitignore file for the Node application. Read more on .gitignore files [here](#).
- ii. **config.json:** Holds the RDS connection credentials/information and application configuration settings (like port and host).
- iii. **package.json:** maintains the project dependency tree; defines project properties, scripts, etc
- iv. **package-lock.json:** saves the exact version of each package in the application dependency tree for installs and maintenance.
- v. **routes.js:** where the code for the API routes' handler functions go.
- vi. **server.js:** the code for the routed HTTP application. It imports routes.js and maps each route function to an API route and type (like GET, POST, etc). It also 'listens' to a specific port on a host using the parameters in config.json

- b. **Server Tests Folder (/server/___tests___):**
 - i. **results.json:** stores (some) expected results for the tests in a json encoding.
 - ii. **tests.js**
 - c. **Client Folder (/client)**
 - i. **.gitignore:** a gitignore file for the client application.
 - ii. **package.json:** maintains the project dependency tree; defines project properties, scripts, etc
 - iii. **package-lock.json:** saves the exact version of each package in the application dependency tree for installs and maintenance
 - d. **Client Public Folder (/client/public):** this folder contains static files like index.html file and other assets for specifying web page titles, crawlability, etc.
 - e. **Client Source Folder (/client/src):** this folder contains the main source code for the React application.
 - i. **config.json:** holds server connection information (like port and host). Could be replaced by a .env file, but students find this easier to manage
 - ii. **fetcher.js:** contains helper functions that wrap calls to API routes. improved testability, reusability, and usability
 - iii. **index.js:** this the main JavaScript entry point to the application and stores the main DOM render call in React. For this application, page routing via components and imports for stylesheets are also embedded in this file.
 - f. **Pages Folder (/client/src/pages):** this folder contains files for React components corresponding to the pages in the application (see the sections below for more details).
 - g. **Component Folder (/client/src/components):** similar to the /pages folder, but this folder contains files for React components corresponding to smaller, reusable components, especially those used by pages.
2. Cd into server using **`cd server`** command on the command line
 3. Install npm package manager using **`npm install`** command.
 4. Cd into the client folder from the root directory using **`cd client`** and repeat step #3.
 5. In the client folder, install the react word cloud package using the command: **`npm install react-cloud --force`**. Please note that the installation will need to be forced given that the package has a dependency on an older version of react (v16).
 6. In both the client and server folders on the command line, run **`npm start`**. The webpage should pop up momentarily after.