REACT JS

Create-react-app is a command that is used to create a **React.js** project with the default configuration. Create-react-app will aid in running React applications

```
node -v
```

```
npm –version
```

If npm and node.js is installed on computer, install create-react-app globally with the command below:

```
npm install –g create-react-app
```

Creation of project − to create a project once above commands are executed, run the below command:

```
npx create-react-app hello-world-example
```

npx comes with npm version 5.2+ . The npm version can be checked on terminal using:

```
npm –version
```

If the npm version is 5.2+, then the react.js project can be directly created with the command:

```
npx create-react-app hello-world-example
```

If npm version is 6+, then execute this command below:

```
npm init react-app hello-world
```

With yarn, we have command:

```
yarn create react-app hello-world-example
```

Once above commands are inserted, change the directory to hello-world-example.

With create-react-app, webpack or babel is required for using advanced features, which are preconfigured. We can only concentrate on writing code.

```
cd hello-world-example
```

To execute the application, run the below command on terminal:

```
npm start
```

npm start runs a live development server, and the code changes will automatically refresh the browser and reflect the changes.

A browser window will be opened, if not opened automatically open a browser window manually and type the URL below:

http://localhost:3000/

3000 is the default port used in the React application. The port number can be changed anytime if there is any issue with the port number, or if it is used by other applications already.

folder structure of a **React** application.

/node_modules: External library files are located in this folder. When we run npm install command again, these files will be overwritten.

/public: Assets/files that aren't compiled or dynamically generated are stored here. These can be static assets like logos or the robots.txt file.

/src: The src or source folder contains all of our React components, external CSS files, etc.

/public folder structure

1) **favicon.ico**:

This the default react icon that always remains in the public folder. We can also put our own project icon here, but the icon extension must be .ico and the icon name may be anything.

We can remove favicon.ico when we place a new favicon for our project/website.

2) **index.html**:

This is the index file that is displayed when the react app opens in a web browser. It contains the HTML template of the react application. index.html file is the root file of the react app. Everything will be rendered through it on the front end. So, don't try to change and remove this file from the public folder.

3) **logo192.png** and **logo512.png**:

These are the logo of react js. It is placed just for the initial view of the react application.

4) **manifest.json**:

manifest.json provides the metadata like short_name, name and icons in the form of JSON for a react application. It may be installed on the mobile or desktop. Then, we can directly open the react application with its installed favicon.

From manifest.json file, users get a notification to install the react application on their mobile or desktop. We can modify the JSON values in this file.

5) **robots.txt**:

The robot.txt file is given just for SEO (Search Engine Optimization) purposes.

/src folder structure

1) **App.css**:

App.css file contains a default CSS code and imports into the App.js file. You can create your own CSS file like App.css. Its name must start with an uppercase letter.

2) **App.js**:

App.js is a parent component file of the react app. It is imported into the index.js file to render content/HTML in the root element that remains in public/HTML.

3) **App.test.js**:

App.test.js creates a testing environment. Basically, it’s written code to protect the react application to from crashing.

We should not modify and remove this file from the react application.

4) **index.css**:

index.css file contains some default css code for index.js. We can modify/add CSS style according to the project.

5) **index.js**:

index.js file is an entry point of the react application. All components renders through this file to the index.html.

6) **logo.svg**:

This is the default logo of react js. We can remove it and place the project logo if needed.

7) **reportWebVitals.js**:

reportWebVital.js is related to the speed of the application.

8) **setupTests.js**:

@testing-library/jest-dom is imported. We should not modify and remove this file from the application.

1) **.gitignore**:

.gitignore file is used to ignore the files that have not to be pushed to the git.

2) **package-lock.json**:

package-lock.json file maintains a version of the installed dependencies.

3) **package.json**:

All the **package/dependencies** are defined in this file. It maintains the dependencies that are necessary for our application.

4) **README.md**:

We can write instructions in this file to configure and set up the react application.