**FOLDERS IN REACT APP**

**Example Project Structure:**

**my-app**

**├── node\_modules**

**├── public**

│ ├── favicon.ico

│ ├── index.html

│ ├── manifest.json

│ └── robots.txt

**├── src**

│ ├── assets

│ │ ├── images

│ │ └── styles

│ │ └── App.css

│ ├── components

│ │ ├── Header.js

│ │ ├── Footer.js

│ │ └── ...

│ ├── pages

│ │ ├── Home.js

│ │ ├── About.js

│ │ └── ...

│ ├── services

│ │ └── api.js

│ ├── App.js

│ ├── index.js

│ └── ...

**├── .gitignore**

**├── package-lock.json**

**├── package.json**

**|── README.md**

The standard structure of a Create React App project includes these essential folders and files. Here’s a detailed breakdown of each:

**Folders and Files:**

1. **node\_modules**:
   * **Description**: Contains all the npm packages and dependencies installed for the project.
   * **Generated by**: npm during the npm install process.
2. **public**:
   * **Description**: Contains the public assets and files that are served directly without any processing by Webpack.
   * **Key Files**:
     + index.html: The main HTML file for your app.
     + favicon.ico: The favicon for your app.
     + manifest.json: Configures the web app's metadata.
3. **src**:
   * **Description**: Contains the source code of your React application.
   * **Key Folders and Files**:
     + **assets**: (Optional) Stores static assets like images and CSS files(styles).
       - images/: Stores image files.
       - styles/: Stores CSS files.
     + **components**: Contains reusable UI components. (Header. Footer).
     + **pages**: Contains components representing different pages/routes. Contains different page components (e.g., Home, About).
     + **services**: (Optional) Contains service files for API calls and business logic.
     + **App.js:** The root component of your application.
     + **index.js:** The entry point of your application.

**App.js**:

import React from 'react';

import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';

import Home from './pages/Home';

import About from './pages/About';

import Header from './components/Header';

import Footer from './components/Footer';

function App() {

return (

<Router>

<Header />

<Switch>

<Route exact path="/" component={Home} />

<Route path="/about" component={About} />

</Switch>

<Footer />

</Router>

);

}

export default App;

**index.js**:

import React from 'react';

import ReactDOM from 'react-dom';

import './assets/styles/App.css';

import App from './App';

ReactDOM.render(

<React.StrictMode>

<App />

</React.StrictMode>,

document.getElementById('root')

);

This structure helps you keep your code modular and organized, making it easier to maintain and scale your application.

1. **.gitignore**:
   * **Description**: Specifies files and directories that should be ignored by Git.
   * **Key Contents**:

node\_modules/

build/

.env

1. **package-lock.json**:
   * **Description**: Automatically generated for any operations where npm modifies either the node\_modules tree or package.json. Ensures that the same versions of packages are installed on every installation.
   * **Generated by**: npm.
2. **package.json**:
   * **Description**: Contains metadata about the project, including dependencies, scripts, and other configurations.
   * **Key Contents**:

{

"name": "my-app",

"version": "0.1.0",

"private": true,

"dependencies": {

"react": "^17.0.2",

"react-dom": "^17.0.2",

"react-scripts": "4.0.3"

},

"scripts": {

"start": "react-scripts start",

"build": "react-scripts build",

"test": "react-scripts test",

"eject": "react-scripts eject"

}

}

1. **README.md**:
   * **Description**: Provides an overview of the project, how to set it up, and how to use it.
   * **Key Contents**:

# My App

This project was bootstrapped with [Create React App](https://github.com/facebook/create-react-app).

## Available Scripts

In the project directory, you can run:

### `npm start`

Runs the app in the development mode.

Open [http://localhost:3000](http://localhost:3000) to view it in the browser.

### `npm test`

Launches the test runner in the interactive watch mode.

### `npm run build`

Builds the app for production to the `build` folder.