📁 Typical React Directory Structure

```
my-app/
├── node_modules/
├── public/
│     └── index.html
├── src/
│     ├── assets/
│     ├── components/
│     ├── pages/
│     ├── App.js
│     ├── App.css
│     ├── index.js
├── .gitignore
├── package.json
├── README.md
```

Let's go through the main folders and files:

✅ public/

This folder contains **static files** that are not processed by Webpack.

File	Description
index.html	The single HTML page React injects into
favicon.ico , logo.png	App icons & assets
Anything in here	Can be accessed directly (e.g. /public/image.png)

✅ src/ (source folder)

This is where **you write all your code**. Everything inside here is compiled by Webpack.

- ◆ `index.js`
  - **Entry point** of your app.
  - React renders the root component (usually `<App />`) here.

    jsx   Copy   Edit

    ```jsx
    import React from 'react'; import ReactDOM from 'react-dom'; import App from './App';
    ReactDOM.render(<App />, document.getElementById('root'));
    ```

- ◆ `App.js`
  - The **main/root component**.
  - It usually holds routing logic, layout, or the base UI structure.

- ◆ `components/`
  - Reusable UI **components** (buttons, cards, navbars, etc.)
  - Example: `Button.js` , `Header.js`

- ◆ `pages/`
  - Full **page-level components** (like Home, About, Contact)
  - Good for larger apps with routing

- ◆ `assets/`
  - Images, icons, CSS files, fonts, etc.

✅ Root-Level Files

File	Description
`package.json`	Lists dependencies, scripts, metadata
`.gitignore`	Files to ignore in Git
`README.md`	Project description and notes
`node_modules/`	Installed packages (auto-generated)

🗂️ Example Layout for a Scalable App

```
src/
├── components/
│      ├── Header.js
│      ├── Footer.js
│      └── Button.js
├── pages/
│      ├── Home.js
│      ├── About.js
│      └── Contact.js
├── assets/
│      ├── images/
│      └── styles/
│             └── App.css
├── App.js
├── index.js
```

🧠 Tips for Managing Your Structure

- **Organize by feature** in bigger apps (e.g. `/features/user/` , `/features/products/`)
- Keep components **small and reusable**
- Group CSS and assets close to the components if they're only used there