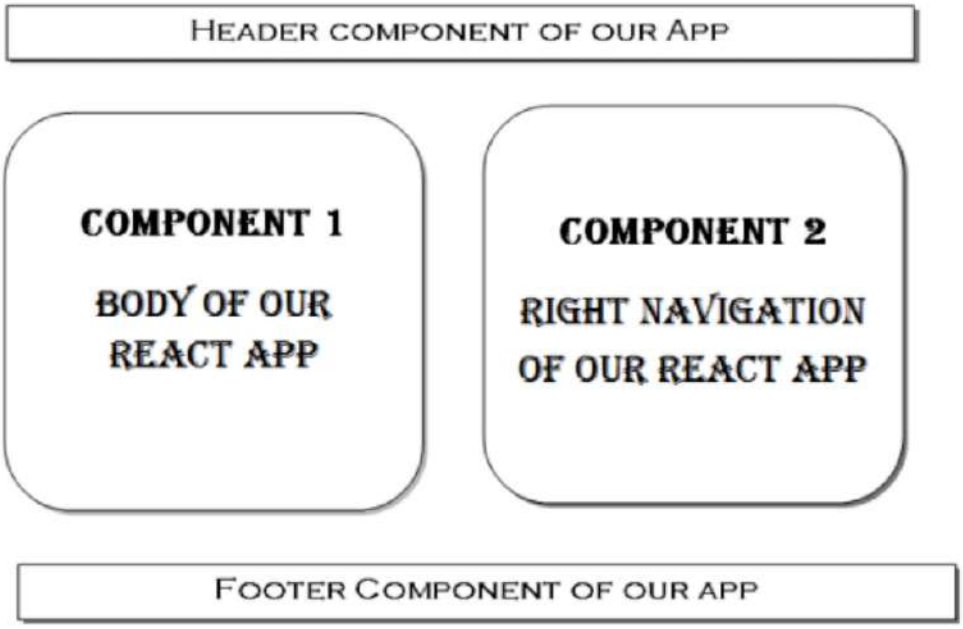
Creating our first react app using create-react-app | Complete React Course in Hindi #2

In the first tutorial, we installed the tools to start working with React. We will now understand the structure of a React app.

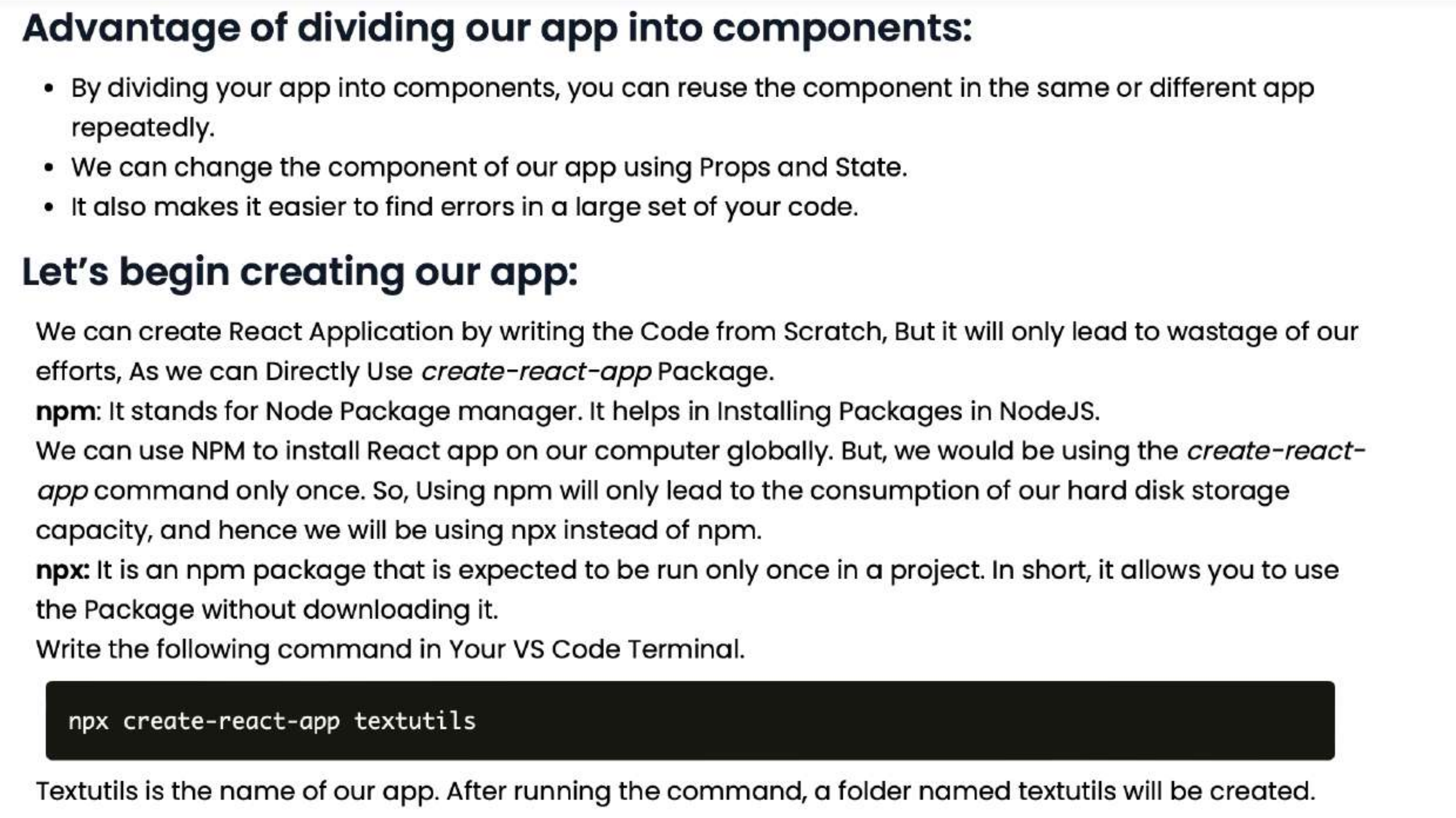
Component-based architecture of React:

In React, we create our app by dividing it into different components. We can create components of our app individually and can reuse them again and again.

For Example: In the below Image, we have structured our app into the header, footer, body, and right navigation bar.


HEADER COMPONENT OF OUR APP
COMPONENT 1
BODY OF OUR REACT APP
COMPONENT 2
RIGHT NAVIGATION OF OUR REACT APP
FOOTER COMPONENT OF OUR APP

You can structure and design your react components as per your choice



Advantage of dividing our app into components:

- By dividing your app into components, you can reuse the component in the same or different app repeatedly.
- We can change the component of our app using Props and State.
- It also makes it easier to find errors in a large set of your code.

Let's begin creating our app:

We can create React Application by writing the Code from Scratch, But it will only lead to wastage of our efforts, As we can Directly Use *create-react-app* Package.
npm: It stands for Node Package manager. It helps in Installing Packages in NodeJS.
We can use NPM to install React app on our computer globally. But, we would be using the *create-react-app* command only once. So, Using npm will only lead to the consumption of our hard disk storage capacity, and hence we will be using npx instead of npm.
npx: It is an npm package that is expected to be run only once in a project. In short, it allows you to use the Package without downloading it.
Write the following command in Your VS Code Terminal.

```
npx create-react-app textutils
```

Textutils is the name of our app. After running the command, a folder named textutils will be created.

Folder Structure of Textutils(Our React App):

1. **node_modules**: It contains all the packages, which are used by React app.
2. **.gitignore**: It contains the files, which you do not want to push in Github.
3. **package.json**: Contains all the packages which are installed in Node modules.
4. **readme.md**: It provides the basic info about your app.

The two folders in which we are interested are :

1. **"public" folder:** It contains an Index.html file**.**

index.html: It is the main HTML file of our react app. This is the page that is displayed on starting our application. It has an empty div tag like this:

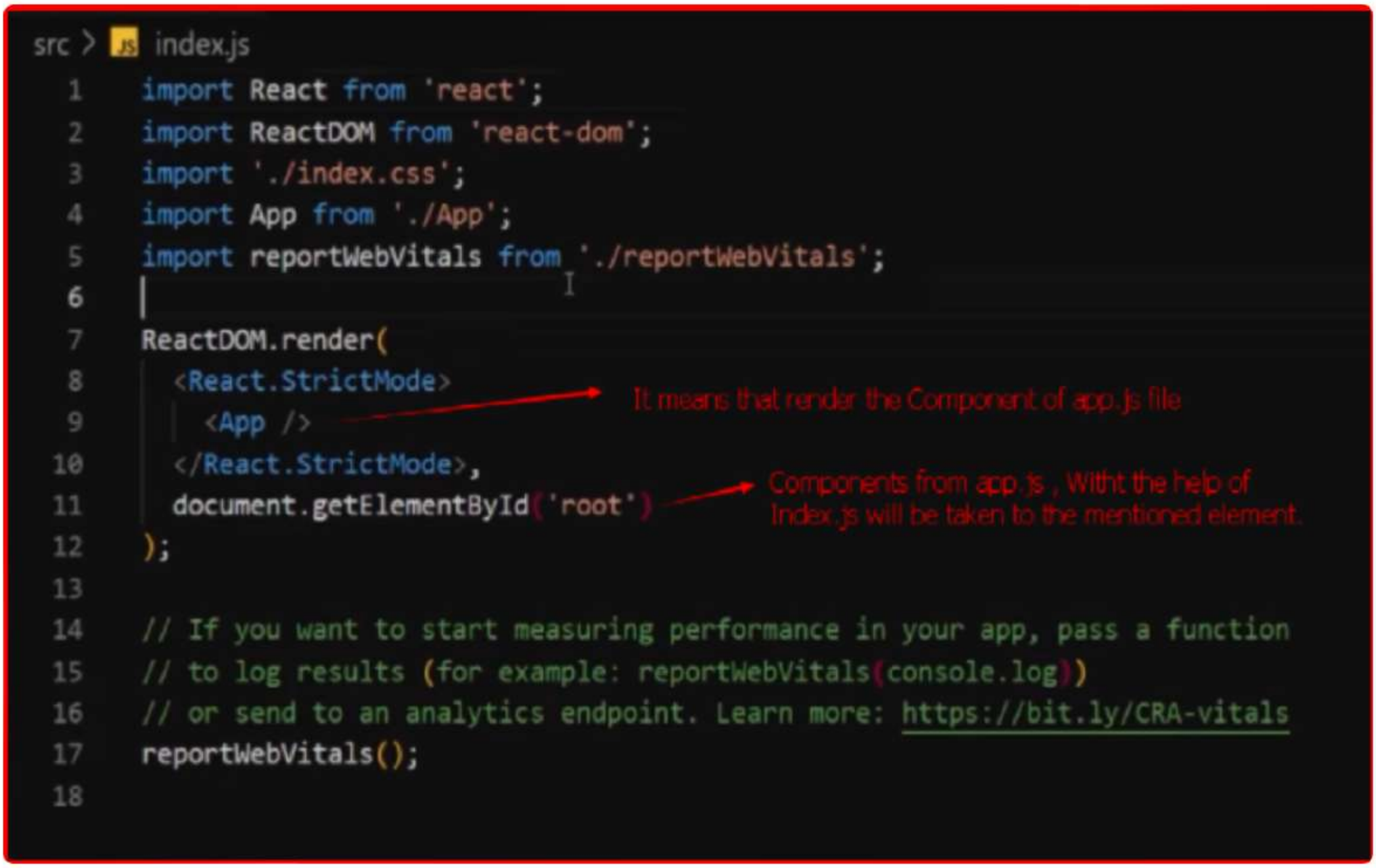
```html
<div id="root"></div>
```

Here, we will be rendering data from components available in the src folder with the help of JavaScript.

- **"src" folder:** Most of the time, we will be writing our code in the src folder.

Two most important files in the src folder are:

- **index.js**: This is the entry component of our app. It has the code of what to render and where.
- **app.js**: With the help of Index.js, the content and changes in it are delivered to the mentioned element.



src > index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

It means that render the Component of app.js file

Components from app.js , Witht the help of Index.js will be taken to the mentioned element.

How to run your React app?

In index.js, open the terminal and write **npm start,** and your react app will be served at *localhost:3000*. You can make changes to the content of your React app in App.js. You can use **npm run build** if you are creating an app for production.