**Introduction to React JS + Installation | Complete React Course in Hindi #1**

React JS is a javascript library originally developed by Facebook. It helps in building highly engaging single-page web applications. React JS helps in breaking down complex UI into simpler components. React JS has a high demand in the field of Web development.

**Installation VS Code, Extensions and Node.js**

**Install VS code**: You can Search ‘Download VS Code’ or directly click here to download VS Code. Open the setup and click next until the ‘Select Additional task’ Dialog box appears. Make sure to check all the options and click next to install the VS Code.

**Install Node.js**: You can Download Node.js by clicking here. Select your OS and click on next, and your Node.js will be installed.

**Download react dev tools**: Click here to add react dev tools chrome extension. It is a chrome extension, which shows your React component hierarchies.

After installation, open the windows power shell (You can do this by pressing Shift + right-click and selecting the windows PowerShell option inside file explorer) and write "node --version" and "npm --version" to check the installation and version of Node.js and Node Package manager (npm) on your computer.

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\iitia> node --version

v16.13.0

PS C:\Users\iitia> npm --version

8.1.0

PS C:\Users\iitia>

**Getting Started with VS code:**

Go to your desired folder and open it inside VS Code. You can do so by right-clicking followed by selecting the option of “Open with Code” in Windows.

**Downloading VS code extensions:**

**Thunderclient**: It is a visual studio code extension that allows you to use 'GET' or 'POST’ requests and test your APIs by easily displaying the response.

**ES7 React/redux/GraphQL/React-native snippets**: It is just like an Emmet abbreviation, which allows you to autocomplete the code.

**Bracket pair colorizer**: It allows matching brackets so that they can be easily identified with colors.

**Auto rename tag**: Automatically rename paired HTML/XML tag.

**Live server**: It allows us to visualize our static page by opening the page with a live server.

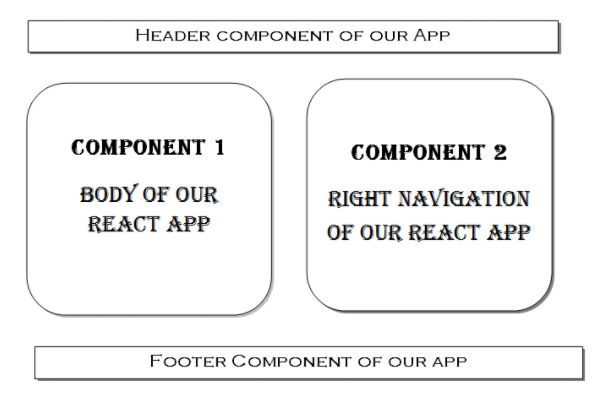
**Creating our first react app using create-react-app | Complete React Course in Hindi #2**

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.



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):**

**node\_modules:** It contains all the packages, which are used by React app.

**.gitignore:** It contains the files, which you do not want to push in Github.

**package.json:** Contains all the packages which are installed in Node modules.

**readme.md:** It provides the basic info about your app.

The two folders in which we are interested are :

**"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:

<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.

**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.