

# Introduction to React

## Topics covered:

- What is ReactJs?
- Why do we use ReactJs?
- What is SPA and MPA?
- What are the Prerequisites to learn ReactJs?
- How to Create a React App?
  - ★ Create-react-app tool
  - ★ What is Npm and Npx?
  - ★ How to run a react project?
- What is the Default structure of Project?
- Let's write our first program in react.

## 1. ReactJs:

React.Js is a JavaScript library for creating interactive and fast user interfaces. It was created by Facebook in 2011 and is now the most widely used JavaScript library for creating SPA and Reusable Components. According to Google, React is the most used JS library for designing user interfaces. React is just concerned with components. A React application is simply a collection of components.

- ReactJs uses JavaScript as it is a JavaScript library.
- Moreover it can use typescript also (strongly typed but similar to javascript).
- React is not a javascript's framework.
- ReactJs is used to make SPA(single page Application).
- ReactJs is used to create Reusable Components.

## 2. Why learn React?

- *Dynamic and Complex UI can be built easily using React* : React is simple to learn and easy to set up. Due to its component architecture, a highly dynamic and complex UI can be built easily and provides good performance.
- *Large Community Support* : React has a large community over Github, It is actively developed and maintained by Facebook and there are too many resources available worldwide.

### 3. SPA(Single Page Application) vs MPA(Multiple Page Application):

#### a. SPA(Single Page Application)

- A Single Page Application is an app that works inside a browser and does not require page reload during use.
- SPAs are all about serving an outstanding user experience (UX) by trying to imitate a “Natural” environment in the browser – no page reloads, no extra wait time.
- It is just one HTML page that you visit which then loads all other content using JavaScript.
- For Example: Gmail, Facebook, Instagram etc.

#### b. MPA(Multiple Page Application):

- A Multi-Page Application works in a traditional way like displaying data, submitting data back to the server, and sending requests to the server to render new pages in the browser.
- Whenever a user navigates from one page to another, a request is sent to the server to send a new HTML file for the URL. The server returns the file and then the HTML file is loaded in the browser.

### 4. Prerequisites:

- Basic Knowledge of HTML and Javascript.
- Basic Programming concepts like: Array, functions, objects, classes etc.
- Some of the ES6 Features.
- Like : Arrow function, Let, Const etc

### 5. How to create a React app:

When we create a react app there are a lot of libraries which are required to build a react application function properly such as React, React-dom, Babel etc.

There are 3 ways to build a react app:

1. Either you can install all the packages required and configure the entire project ourselves.
2. We can use the tool officially provided to build a react app and develop it accordingly.
3. Use an online code playground. Try CodeSandbox (codesandbox.io).  
how to use it - Navigate to the link; this will create the React project from scratch. In the center window, start making the changes in App.js. The changes will immediately reflect in the right window.

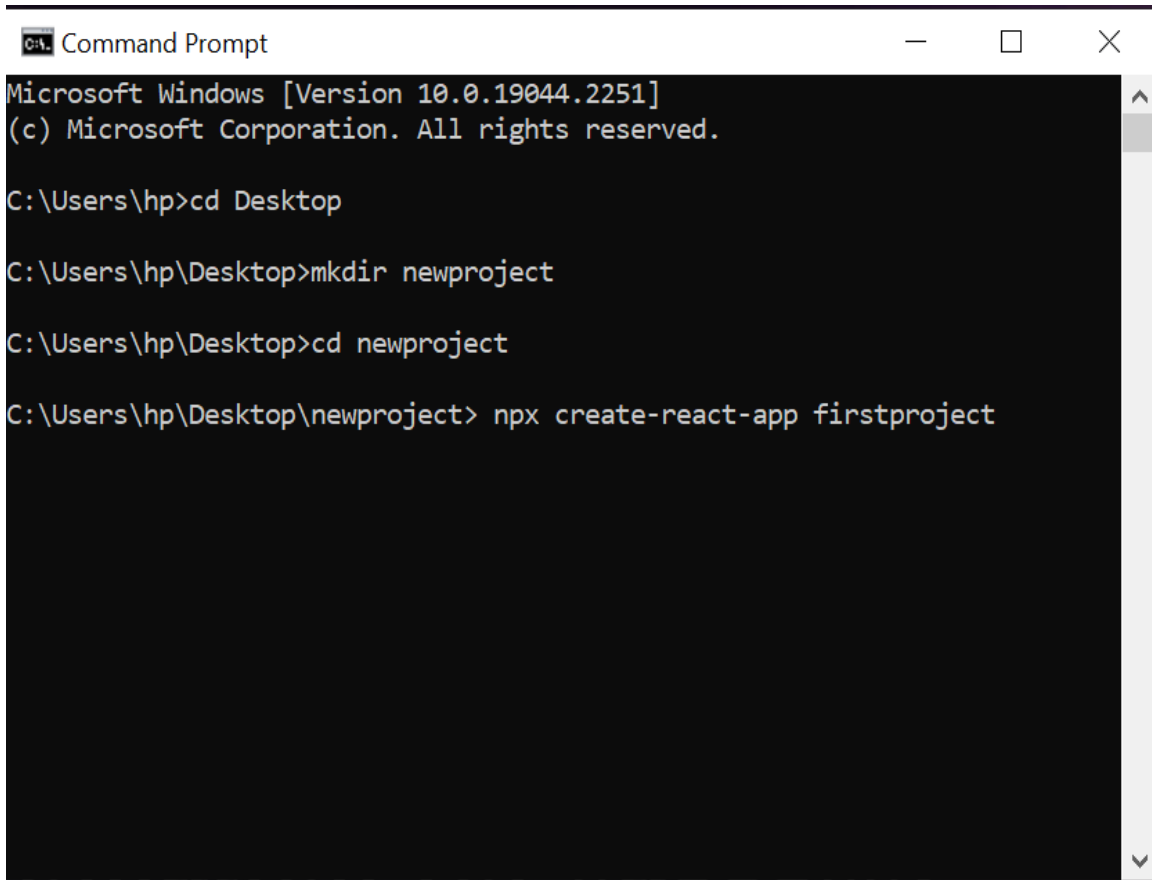
We are gonna use the second one:

**`npx create-react-app firstproject`**

Must remember few things:

- Node must be there in your device.  
To check open the command prompt and check `node -v` and `npm -v`.
- If node is not downloaded : download it from - <https://nodejs.org/en/download/>
- The name of the project must be in small letters.
- Even the whole command must be in small letters.
- Write the command in the command prompt where you want to build a new react project template. (path must be accurate).

Just write this command in your terminal, like this:



```
Command Prompt
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>cd Desktop

C:\Users\hp\Desktop>mkdir newproject

C:\Users\hp\Desktop>cd newproject

C:\Users\hp\Desktop\newproject> npx create-react-app firstproject
```

```
C:\Users\hp\Desktop>npm install react react-dom react-scripts cra-template

C:\Users\hp\Desktop>mkdir newproject

C:\Users\hp\Desktop>cd newproject

C:\Users\hp\Desktop\newproject>npx create-react-app firstproject
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

Creating a new React app in C:\Users\hp\Desktop\newproject\firstproject.

npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
[ ] | idealTree:@babel/generator: sill placeDep ROOT @j
```

**a. Create-react-app Tool**

- The create-react-app Tool Installs all the dependencies and packages required to build a react application.
- It creates a default configuration for our React Project.
- It also adds some starter files in the newly created app.

**b. Npm and Npx**

- Npm stands for node package manager.
- It helps to manage third party packages and dependencies that you will be installing you application.
- It is installed automatically when you install Node on your device.
- Npx is the node package executor or runner.
- It is used to download and run packages temporarily.
- Here we are using npx to create a react app because we will be using create react app once to just create an application so we don't need this package permanently in our project.

**c. Let's see how to run a react application:**

Write this command on the terminal:

**npm start**

(Before running the above command you must change your directory to your project directory as shown in the image)

```
Command Prompt

npm run build
  Bundles the app into static files for production.

npm test
  Starts the test runner.

npm run eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back
  !

We suggest that you begin by typing:

  cd firstproject
  npm start

Happy hacking!

C:\Users\hp\Desktop\newproject>cd firstproject
C:\Users\hp\Desktop\newproject\firstproject>_
```

```
Command Prompt

npm run build
  Bundles the app into static files for production.

npm test
  Starts the test runner.

npm run eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back
  !

We suggest that you begin by typing:

  cd firstproject
  npm start

Happy hacking!

C:\Users\hp\Desktop\newproject>cd firstproject
C:\Users\hp\Desktop\newproject\firstproject>npm start_
```

```
Windows PowerShell

Compiled successfully!

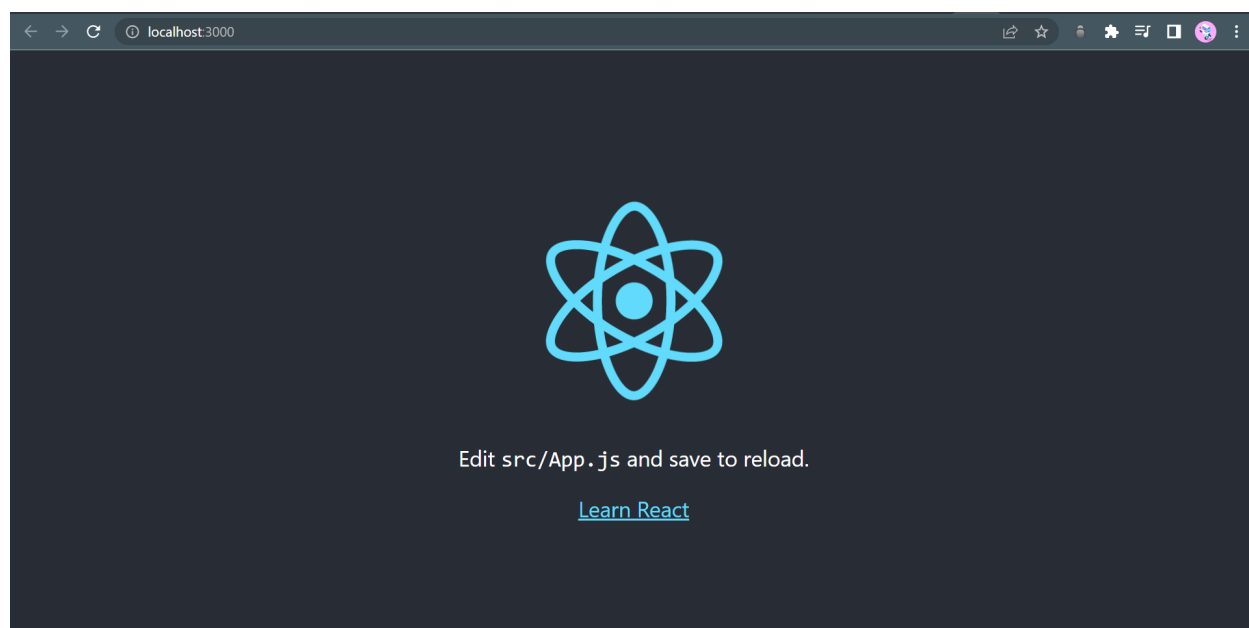
You can now view firstproject in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Server will start at 3000 port and default page will show:



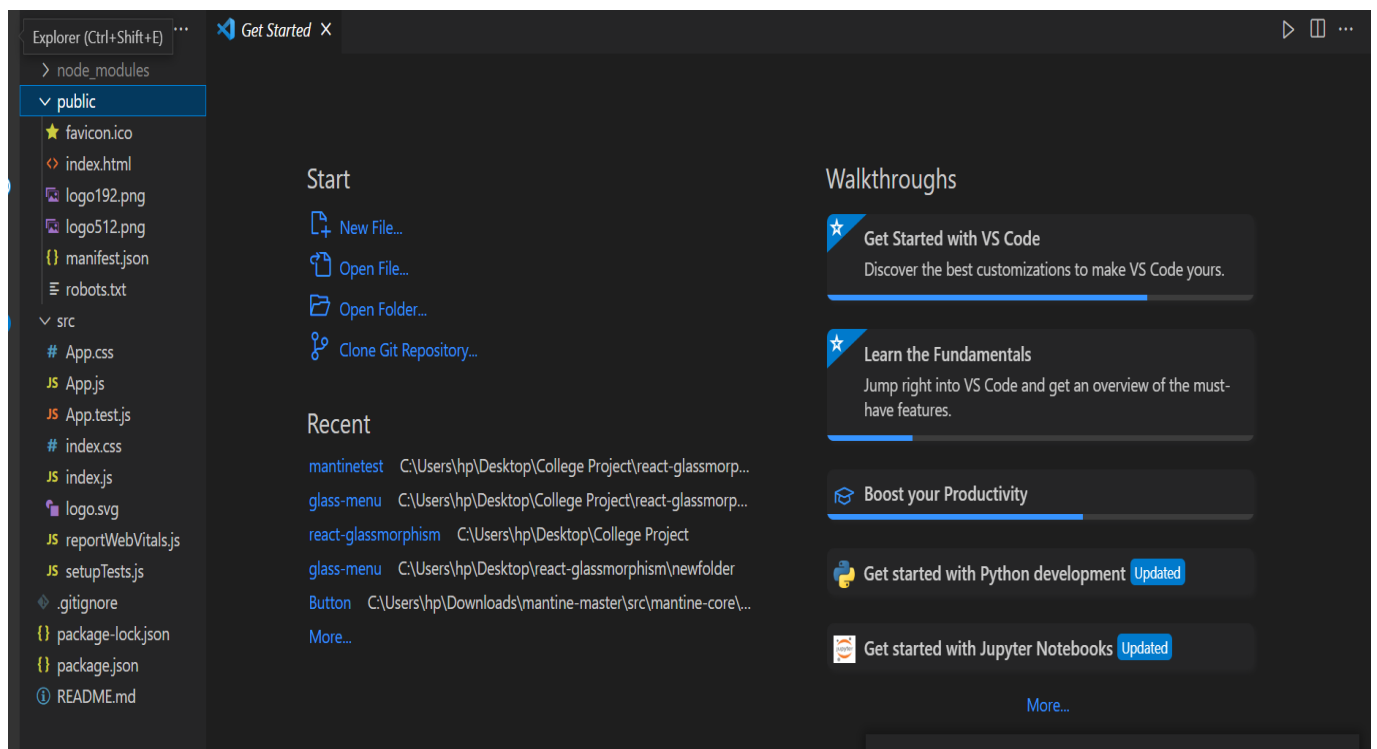
## 6. Default Structure of React project Folder:

When you look at the project structure, you'll notice a `/public` and `/src` directory, as well as the standard node modules, `.gitignore`, `README.md`, and `package.json` files.

Our important file in `/public` is `index.html`, which is quite identical to the static `index.html` file we created before — just a root `div`.

No libraries or scripts are being loaded at this time. All of our React code will be stored in the `/src` directory.

Find the following line in `/src/App.js` to show how the environment automatically compiles and updates your React code:



## 7. Let's write our first program in react:

**App.js:**

```
JS App.js  X
src > JS App.js > App
1  import './App.css';
2
3  function App() {
4    return (
5      <div className="App">
6        <h1>Hello React</h1>
7      </div>
8    );
9  }
10
11  export default App;
12
```

In the result of the App function that was returned above, take note of how we wrote what appears to be HTML.

This is not HTML nor JavaScript, nor is it React. It's called JSX. It's a JavaScript extension that allows us to write function calls in HTML-like syntax.

Save this and check your browser; it should look like this -

