**MODULE: 3 (JavaScript Essentials)**

**1. What is React Js?**

**Ans.** The React.js framework is an open-source JavaScript framework and library developed by Facebook. It’s used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript.

In React, you develop your applications by creating reusable components that you can think of as independent Lego blocks. These components are individual pieces of a final interface, which, when assembled, form the application’s entire user interface.

React’s primary role in an application is to handle the view layer of that application just like the V in a model-view-controller (MVC) pattern by providing the best and most efficient rendering execution. Rather than dealing with the whole user interface as a single unit, React.js encourages developers to separate these complex UIs into individual reusable components that form the building blocks of the whole UI. In doing so, the ReactJS framework combines the speed and efficiency of JavaScript with a more efficient method of manipulating the DOM to render web pages faster and create highly dynamic and responsive web applications.

**2. What is NPM in React Js?**

**Ans.** NPM is short for *node package manager*, an online directory that contains the various already registered open-source packages. NPM modules consume the various functions as a third-party package when installed into an app using the NPM command npm install.

Third-party NPM modules are the unit of logic for a specific functionality or a full-fledged library, such as *reactstrap*, *material-ui*, *redux-form*, and so on. In this guide, you will learn how to load NPM modules in your app using the NPM command or the GitHub source code.

Load the NPM Module Using the CLI

Any NPM modules can be easily consumed in your app using the CLI command followed by the module name. Below is the syntax used to install the NPM module.

1npm install <module\_name>

shell

Try installing any module using the above CLI command. Let's install *reactstrap* for an example.

1npm install reactstrap

shell

When you execute the above command, the related module folder will get added to the directory node\_modules in your current app, and you will be able to use the components or functionality of the module anywhere in your app.

The installed package is reactstrap, but you may wonder how to use it in the component. Import the module as given below.

1import React, { Component } from "react";

2// Import the module

3import { Button, Popover, PopoverHeader, PopoverBody } from "reactstrap";

4

5class SimplePopover extends Component {

6 render() {

7 const { popoverOpen } = this.state;

8

9 return (

10 <div>

11 // Using the module's component

12 <Popover

13 placement="bottom"

14 isOpen={popoverOpen}

15 target="mypopover"

16 toggle={this.togglePopover}

17 >

18 <PopoverHeader>This is popover title</PopoverHeader>

19 <PopoverBody>

20 This is simple popover content

21 </PopoverBody>

22 </Popover>

23 </div>

24 );

25 }

26}

27export default SimplePopover;

jsx

You will notice in the above code that there is one statement that imports the components from the installed module.

1 import { Button, Popover, PopoverHeader, PopoverBody } from "reactstrap";

jsx

You will be able to use it into the components after importing the required component from the module.

Load the NPM Module Using the GitHub Repository

You can use any NPM module by installing it using the CLI command, but alternatively, it can also be used directly by providing the GitHub repository URL as shown below.

1npm install <git\_repo\_url>

shell

Along with the CLI command, add the required GitHub repository URL of the module that you want to install.

1npm install https://github.com/reactstrap/reactstrap.git

shell

A folder gets created inside the directory node\_modules by providing the GitHub URL along with the install command. And the package gets mentioned in package.json.

Load the NPM Module Using the package.json File

The file package.json contains all the installed dependencies' details, such as module and installed version. Still, if you want to install the module directly, you can do it by mentioning the two parameters, as shown below.

1 {

2 "dependencies": {

3 "package/module name": "version"

4 }

5}

json

Specify the module you want to install and its required version so once the command npm install is executed, the mentioned package will be installed with all other packages.

So, in that case, npm install package\_name does not require installing the package directly. This approach is quite handy when you know all the needed packages and versions.

Conclusion

Most JavaScript-based libraries or frameworks require the NPM packages to go ahead with app development. Hence the packages should be installed using one of the ways explained in this guide.

**3.** **What is Role of Node Js in react Js?**

**Ans.**

| **Node.js** | **React.js** |
| --- | --- |
| Node.js used as a back-end framework | React is used for developing user interfaces. |
| It supports the Model–view–controller (MVC) framework. | Does not support the Model–view–controller (MVC) framework. |
| It runs on chrome’s v8 engine and uses an event-driven, non-blocking I/O model, which is written in C++. | It uses Node.js to compile and optimize the JavaScript code and easy to create UI Test cases. |
| Node.js handles requests and authentication from the browser, make database calls, etc. | It makes API calls and processes in-browser data. |
| Here the Real-time data streaming is handled easily. | In React complex architecture makes it hard to keep track of the traditional approach. |
| Framework for JavaScript execution having the largest ecosystem of open source libraries. | Facebook-backed Open Source JS library. |
| The language used is only JavaScript. | The language used is JSX and JavaScript. |
| There is no DOM (Document Object Model) concept that is Used. | Here the Virtual DOM (Document Object Model) is Used that makes it faster. |
| It is easy to write microservices in Node.Js | Microservices are difficult to be written in React.Js |
| It is highly scalable. | Scalability is still a challenge. |
| It has a simple architecture. | It has a complex architecture. |

**4. What is CLI command In React Js?**

**Ans.** ReactJS is a simple, feature rich, component based JavaScript UI library. It can be used to develop small applications as well as big, complex applications. ReactJS provides minimal and solid feature set to kick-start a web application. React community compliments React library by providing large set of ready-made components to develop web application in a record time. React community also provides advanced concept like state management, routing, etc., on top of the React library.

## React versions

The initial version, 0.3.0 of React is released on May, 2013 and the latest version, *17.0.1* is released on October, 2020. The major version introduces breaking changes and the minor version introduces new feature without breaking the existing functionality. Bug fixes are released as and when necessary. React follows the *Sematic Versioning (semver)* principle.

## Features

The salient features of *React library* are as follows −

* Solid base architecture
* Extensible architecture
* Component based library
* JSX based design architecture
* Declarative UI library

## Benefits

Few benefits of using *React library* are as follows −

* Easy to learn
* Easy to adept in modern as well as legacy application
* Faster way to code a functionality
* Availability of large number of ready-made component
* Large and active community

## Applications

Few popular websites powered by *React library* are listed below −

* *Facebook*, popular social media application
* *Instagram*, popular photo sharing application
* *Netflix*, popular media streaming application
* *Code Academy*, popular online training application
* *Reddit*, popular content sharing application

As you see, most popular application in every field is being developed by *React Library*.

**5. What is Components in React Js?**

**Ans.** Earlier, the developers write more than thousands of lines of code for developing a single page application. These applications follow the traditional DOM structure, and making changes in them was a very challenging task. If any mistake found, it manually searches the entire application and update accordingly. The component-based approach was introduced to overcome an issue. In this approach, the entire application is divided into a small logical group of code, which is known as components.

A Component is considered as the core building blocks of a React application. It makes the task of building UIs much easier. Each component exists in the same space, but they work independently from one another and merge all in a parent component, which will be the final UI of your application.

Every React component have their own structure, methods as well as APIs. They can be reusable as per your need. For better understanding, consider the entire UI as a tree. Here, the root is the starting component, and each of the other pieces becomes branches, which are further divided into sub-branches.

**6. What is Header and Content Components in React Js?**

**Ans.** In React JS, a component is a modular and reusable piece of code that encapsulates functionality and data. There are two main types of components in React JS: functional components and class components.

When it comes to building web applications with React, the header and content components are commonly used to structure the page layout.

The Header component typically includes the website's logo, navigation menu, and any other information that is displayed at the top of the page. It usually remains static and is visible on every page of the website.

The Content component contains the main content of the page, such as text, images, forms, and other UI elements. The content of this component changes dynamically based on user interactions, data from APIs, or other sources.

To build these components in React, you can create a separate file for each component and then import and render them in your main application file. This approach allows you to reuse the same components across multiple pages of your website and makes your code more modular and easy to maintain.

**7. How to install React Js on Windows, Linux Operating System? How to install NPM and How to check version of NPM?**

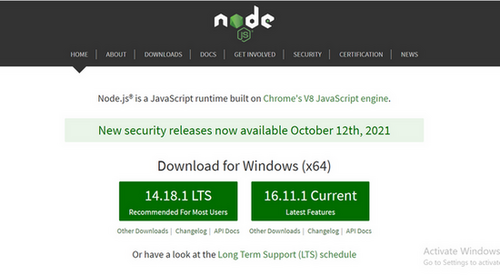
**Ans. React JS** -React is an open-source component-based front-end JavaScript library. It is used to create fast and interactive user interfaces for web and mobile applications. It is easy to create a dynamic application in React because it requires less coding and offer more functionality. It is used by big MNC and fresh new startups

### **Features of React:**

1. **Reusable Components**: A single React app consists of many components each component have their own logic and code  but we can easily reuse components any number of time hence reducing the developers time and increasing the efficiency of work
2. **Debugging**: React app can be easily debug using  “react developer tools”.It’s a browser extension that can be used for both chrome as well as Firefox.

### **Installation Reactjs on Windows:**

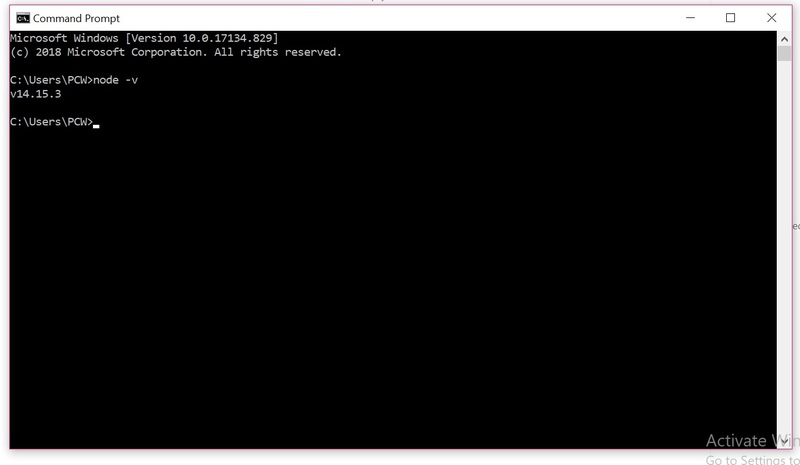
**Step 1**: Install Node.js installer for windows. Click on this [link](https://nodejs.org/en/). Here install the LTS version (the one present on the left). Once downloaded open NodeJS without disturbing other settings, click on the **Next**button until it’s completely installed.



*Install the 14.18.1 LTS*

**Step 2**: Open command prompt  to check whether it is completely installed or not type the command –>

node -v

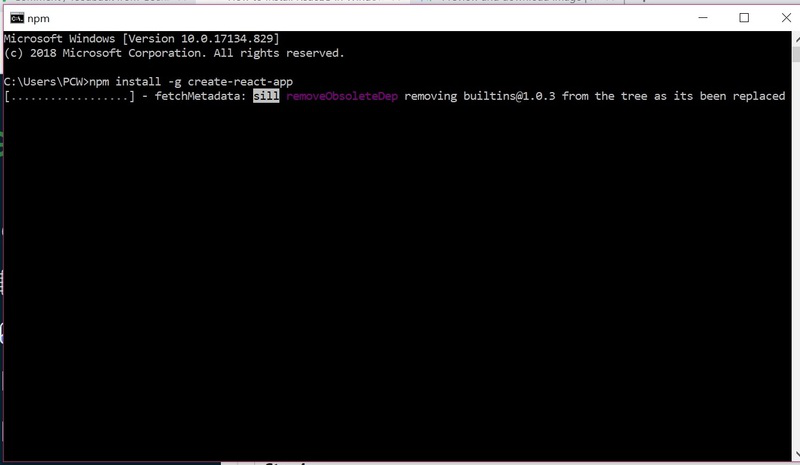


*Node Version is v14.15.3*

If the installation went well it will give you the version you have installed

**Step 3**: Now in the terminal run the below command:

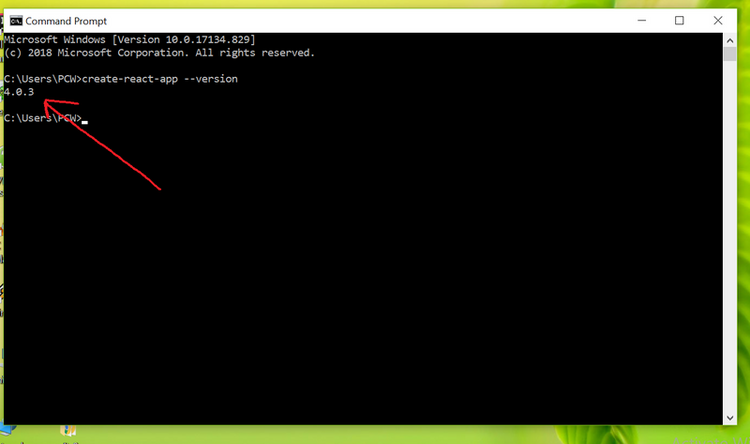
npm install -g create-react-app



*Installation will take few seconds*

It will globally install react app for you. To check everything  went well run the command

create-react-app --version



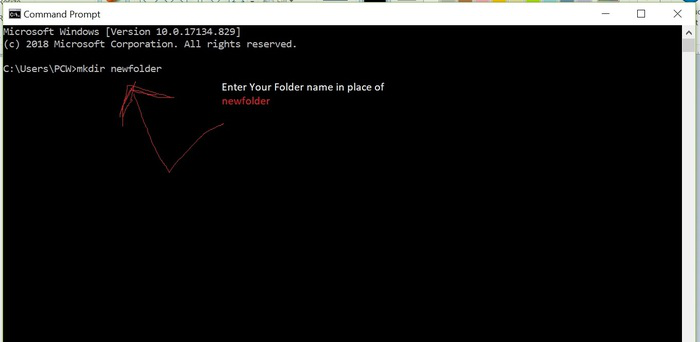
*version  4.0.3*

If everything went well it will give you the installed version of react app

**Step 4:**Now Create a new folder where you want to make your react app using the below command:

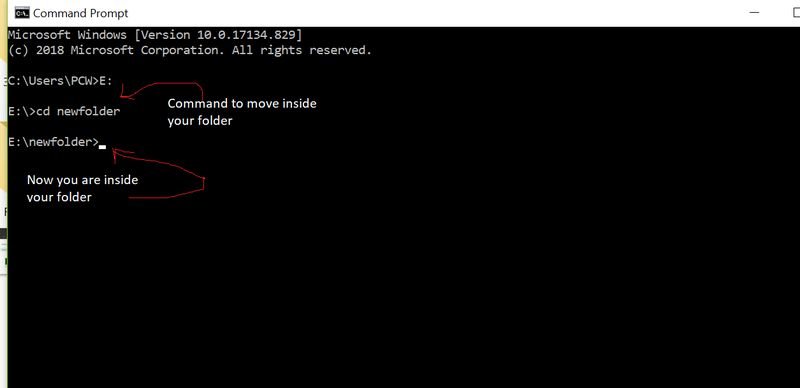
mkdir newfolder

**Note:** The *newfolder*in the above command is the name of the folder and can be anything.



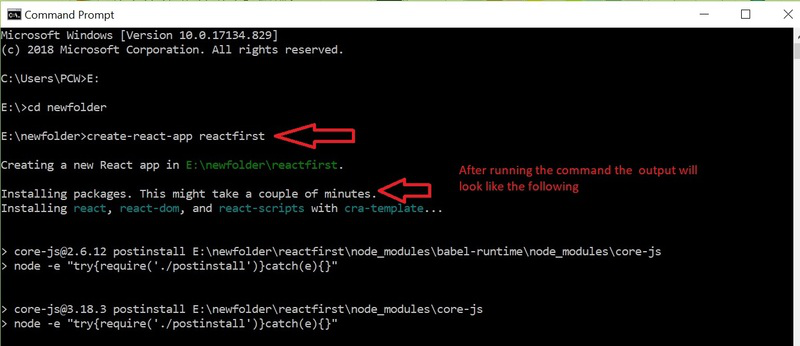
Move inside the same folder using the below command:

cd newfolder (your folder name)



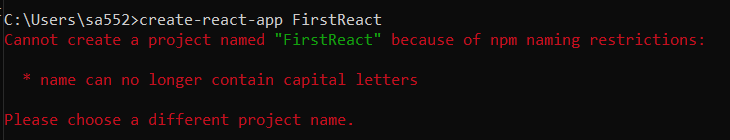
**Step 5**: Now inside this folder run the command –>

create-react-app reactfirst YOUR\_APP\_NAME

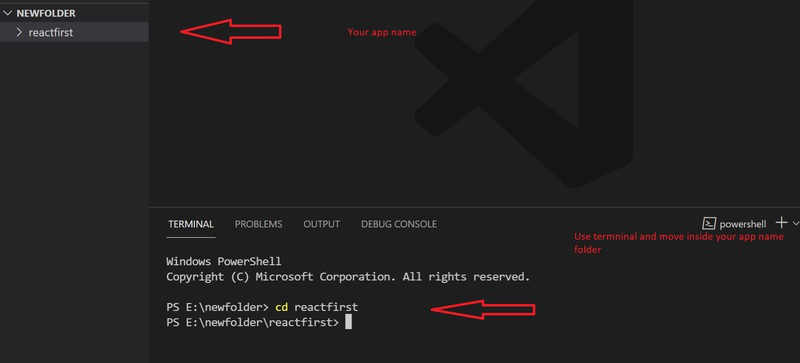


It will take some time to install the required dependencies

**NOTE:**Due to npm naming restrictions, names can no longer contain capital letters, thus type your app’s name in lowercase.

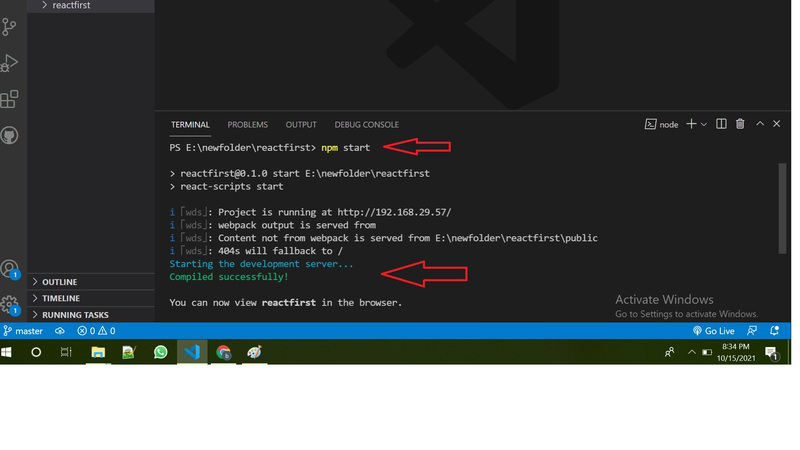


**Step 6**: Now open the IDE of your choice for eg.  Visual studio code and open the folder where you have installed the react app **newfolder**(in the above example)  inside the folder you will see your app’s name **reactapp**(In our example). Use the terminal and move inside your app name folder.Use command  **cd reactapp**(your app name)



**Step 7:**To start your app run the below command :

npm start



Once you run the above command a new tab will open in your browser showing React logo as shown below :



Congratulation you have successfully installed the react-app and are ready to build awesome websites and app

**8. How to check version of React Js?**

**Ans.**

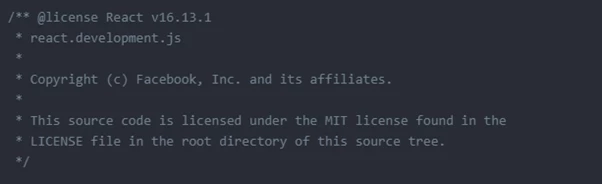
Various methods are available to check the installed ReactJS version on your [operating system](https://monovm.com/blog/what-is-an-operating-system/). Here we have added some of the most commonly used methods to check React version; what current react version do you have?

**Methode 1:** You can check React JS version directly by visiting the ***JSON*** file. You can find the version of the React app under the dependencies{} section, as shown below.



As per the output, you can see the ReactJS version as 16.13.1.

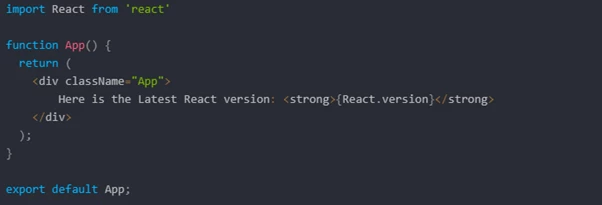
**Method 2:** Another simple method to check React latest version is by navigating to the ***node\_modules/react/cjs/react.development.js***You can check the version under the comment section below the image.



As per the output, you can see the ReactJS version as 16.13.1.

**Method 3:** You can use the following method to check the React JS version that you are using on your system. You can add the below-mentioned code in your React component, and it will display the result using the [web browser](https://monovm.com/blog/what-is-a-web-browser/).

Replace the following mentioned code with the existing code in the ***src/App.js***

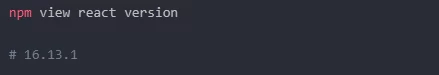


After changing the code, save the content of the file. Now, you can run the below command to display the React version on the web browser.

npm start

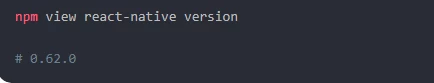
**Method 4:** Use the command line to check the latest version of ReactJS. You can run the below command.

npm view react version



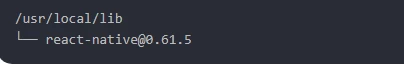
**Method 5:** You can run the following command to check the version of the React native.

npm view react-native version



**Method 6:** You can use the below-mentioned command to check the react-native version installed on your local development system command.

npm ls react-native -g



**9. How to change in components of React Js?**

**Ans.** To change the state of the React component is useful when you are working on a single page application, it simply replaces the content of the existing component for the user without reloading the webpage.

We have to set initial state value inside constructor function and set click event handler of the element upon which click, results in changing state. Then pass the function to the click handler and change the state of the component inside the function using **setState**. The **setState** function used to change the stateof the component directly or with the callback approach as mentioned below.

**Example 1:**

**index.js:**

* Javascript

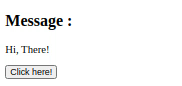
|  |
| --- |
| import React from 'react'  import ReactDOM from 'react-dom'  import App from './App'    ReactDOM.render(<App />, document.querySelector('#root')) |

**App.js:**

* Javascript

|  |
| --- |
| import React, { Component } from 'react'    class App extends Component {    constructor(props){  **super**(props)        // Set initial state  **this**.state = {msg : 'Hi, There!'}        // Binding this keyword  **this**.handleClick = **this**.handleClick.bind(**this**)    }      handleClick(){        // Changing state  **this**.setState({msg : 'Welcome to the React world!'})    }      render(){  **return** (        <div>          <h2>Message :</h2>      <p>{**this**.state.msg}</p>                {/\* Set click handler \*/}          <button onClick={**this**.handleClick}>            Click here!          </button>        </div>      )    }  }    export **default** App |

**Output:**



**10. How to Create a List View in React Js?**

**Ans.**

**Input:**

**ChekBoxList.js**

import React from 'react'

import styles from './my-style.module.css'

function ChekboxList() {

  return (

    <div>

       {/\* chekbox start function \*/}

      <div className="col-md-5 mt-5 offset-4">

      <h1>The "React way" to Reander a list</h1>

      </div>

      <div className="col-md-3 mt-5 offset-4">

        <div className="input-group">

          <div className="input-group-text">

            <input className="form-check-input mt-0" type="checkbox" defaultValue aria-label="Checkbox for following text input" />

          </div>

          <input type="text" className="form-control" aria-label="Text input with checkbox" />

        </div>

      </div>

      <div className="col-md-3 mt-5 offset-4">

        <div className="input-group">

          <div className="input-group-text">

            <input className="form-check-input mt-0" type="checkbox" defaultValue aria-label="Checkbox for following text input" />

          </div>

          <input type="text" className="form-control" aria-label="Text input with checkbox" />

        </div>

      </div>

      <div className="col-md-3 mt-5 offset-4">

        <div className="input-group">

          <div className="input-group-text">

            <input className="form-check-input mt-0" type="checkbox" defaultValue aria-label="Checkbox for following text input" />

          </div>

          <input type="text" className="form-control" aria-label="Text input with checkbox" />

        </div>

      </div>

      <div className="col-md-3 mt-5 offset-4">

        <div className="input-group">

          <div className="input-group-text">

            <input className="form-check-input mt-0" type="checkbox" defaultValue aria-label="Checkbox for following text input" />

          </div>

          <input type="text" className="form-control" aria-label="Text input with checkbox" />

          {/\* chekbox end function \*/}

        </div>

      </div>

    </div>

  )

}

export default ChekboxList

**my-style.module.css**

div{

    background-color: rgb(115, 222, 115);

}

**App.js**

import React from "react";

import ChekBoxList from "./ChekBoxList"

function App() {

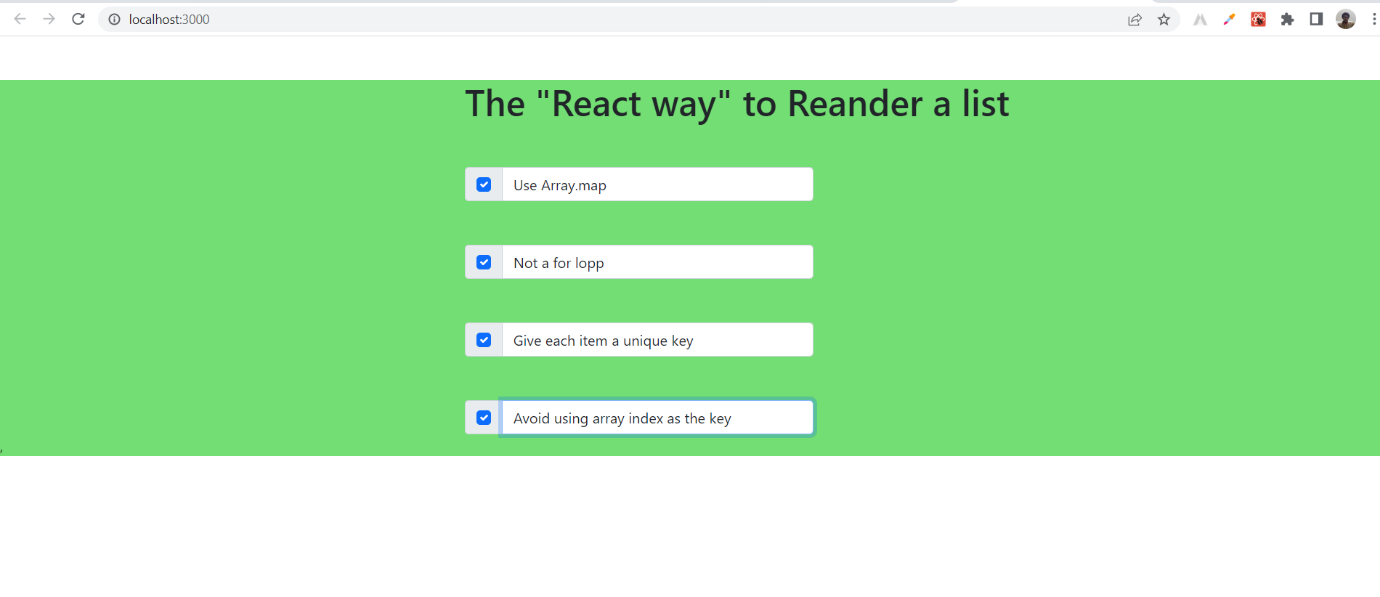
  return (

      <ChekBoxList />

  );

}

export default App;

**Output:**

**11. Create Increment decrement state change by button click?**

**Ans.**

**App.js**

import React, {useState} from "react";

function App() {

  //Increment decrement function

  const [count, setCount] = useState(0);

  return (

    <>

   <div className="container my-5">

    <div className="card text-center my-5">

      <div className="card-body">

          <h1>Counter app</h1>

          <div className="my-5">

            <h2 className="my-5">{count}</h2>

            {/\* button start \*/}

            <button className="btn btn-success mx-3" onClick={() => setCount(count + 1)}>Increment</button>

            <button className="btn btn-danger mx-3" onClick={() => setCount(count - 1)} disabled={count === 0}>Decrement</button>

            <button className="btn btn-secondary mx-3" onClick={() => setCount(0)} disabled={count === 0}>Reset</button>

            {/\* button end \*/}

          </div>

      </div>

    </div>

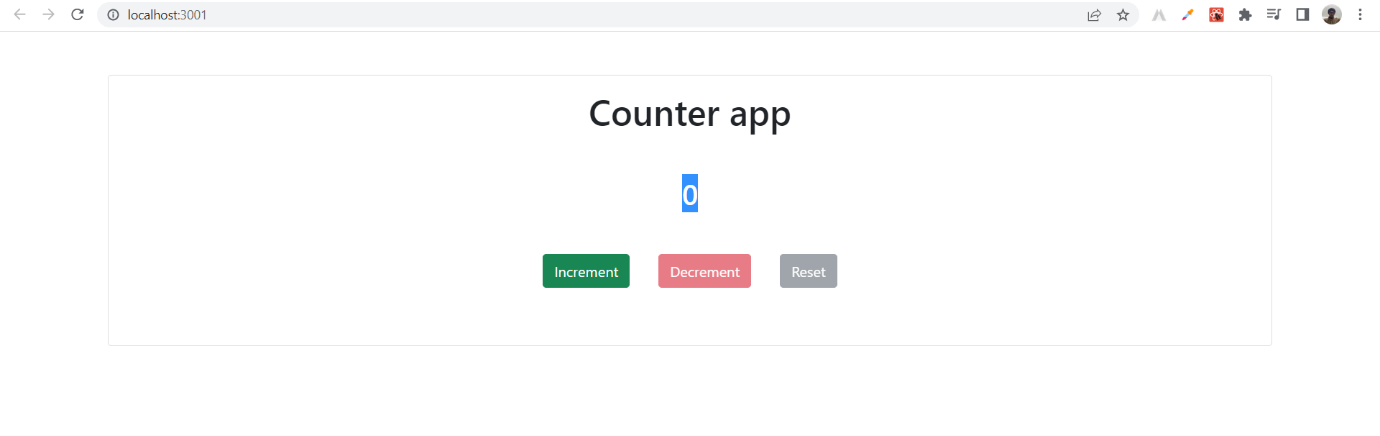
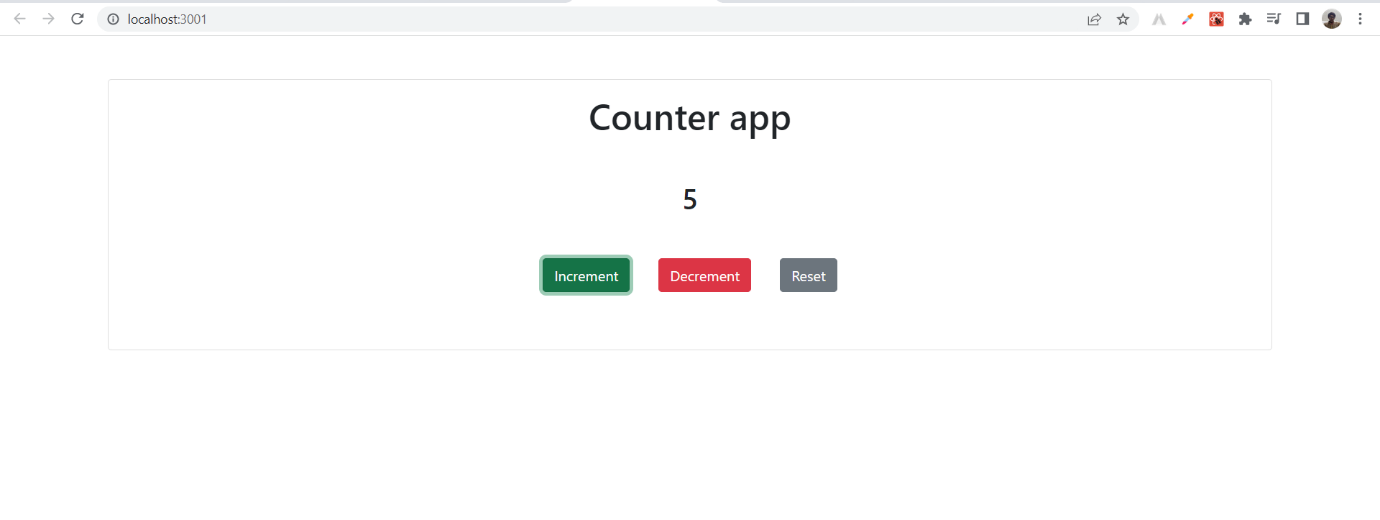
   </div>

   </>

  );

}

export default App;

**Output. **

Top of Form