REACT JS:::

**NEED :**

1:Nodejs

2:Reactjs

3:Visual studio code

**1:Nodejs install** : https://nodejs.org/en/download/

C:\Users\DELL\AppData>node --version

v16.13.0

**2: Reactjs install**: npm install -g create-react-app (eg: E:\reactjs>npm install -g create-react-app)

create -react -app -v

Three(3) commands for creating ReactJS APP::

1:npm install create-react-app

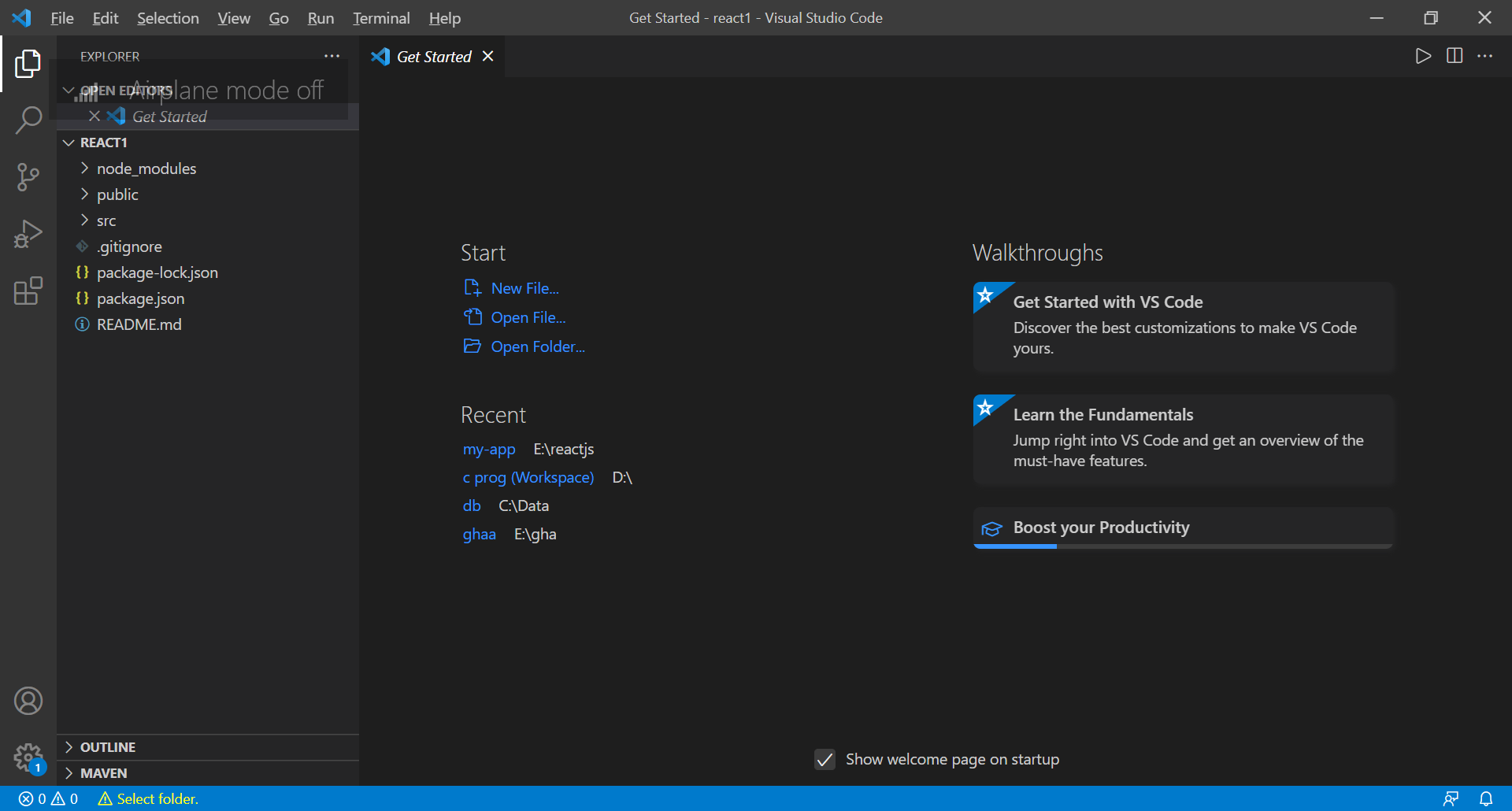
2:npx create-react-app project\_name (project\_name can be any name)

E:\reactjs>npx create-react-app react1

3:npm start(displays when above command is success)

\*E:\reactjs>cd react1 (Move to the project directory)

\*E:\reactjs\react1>code . ((code .) -when it is used then directly visual studio is opened)



\*Public folder -> index file -> Which run on browser.

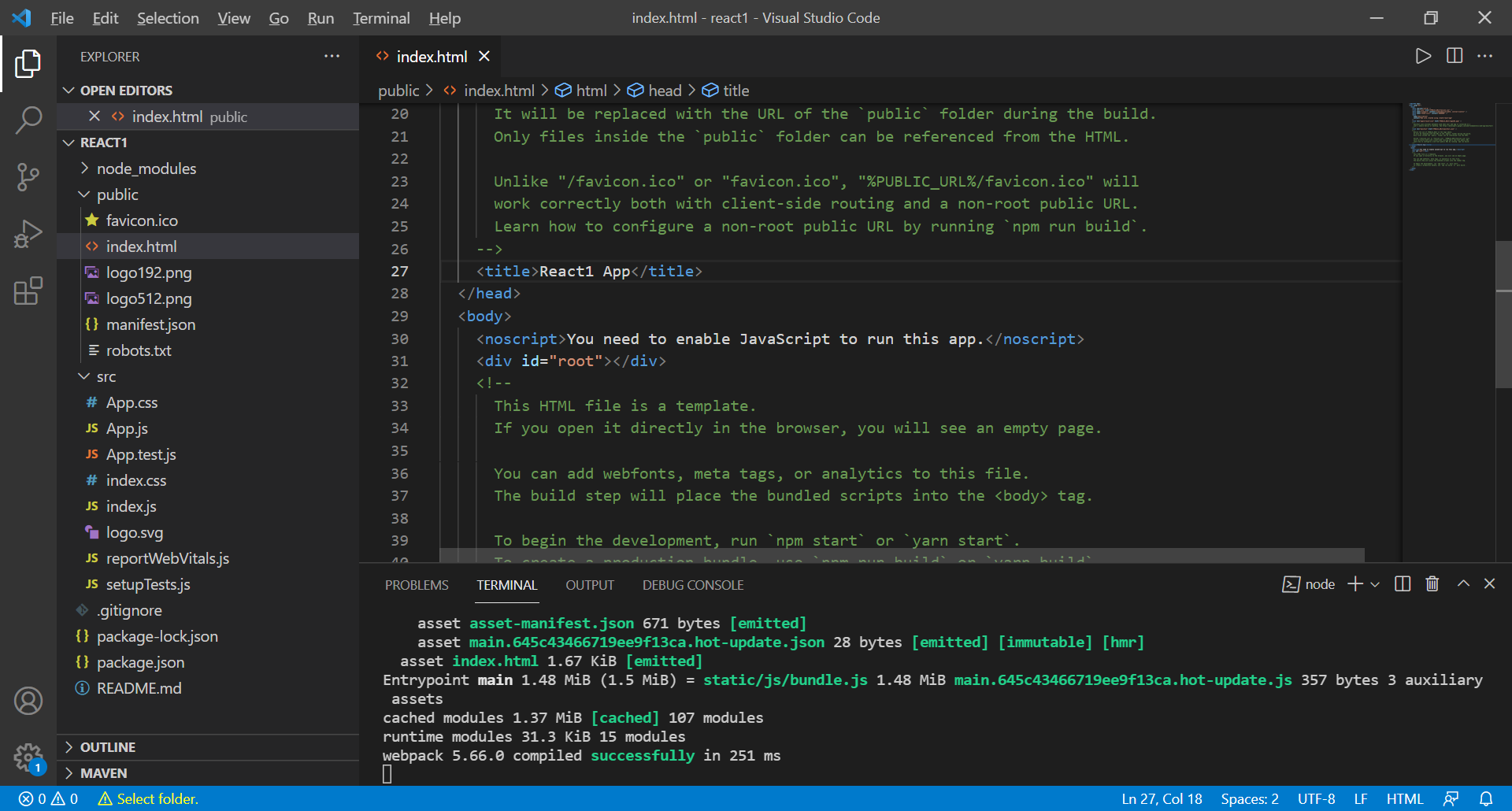
\*src -> components creation or modifications.

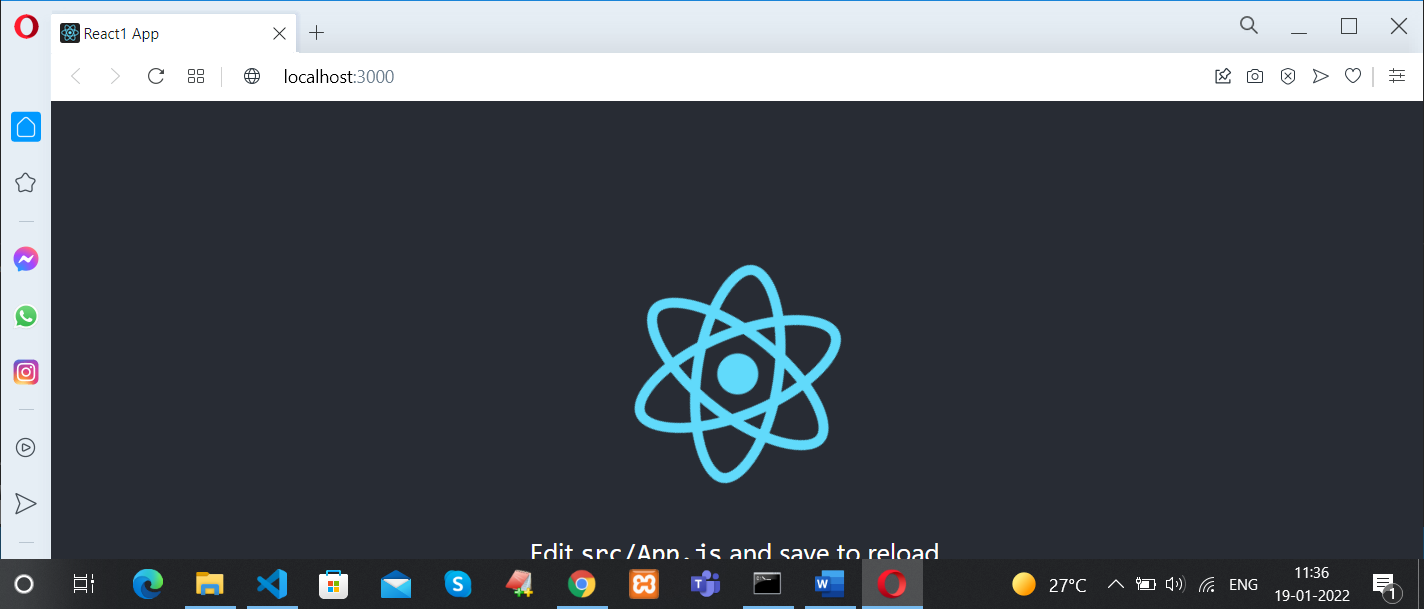
\***HOW TO START APPLICATION IN VISUAL:::**

Terminal-> New Terminal (src ->App.js ->it runs)->npm start🡪(Starts in server and opens browser)

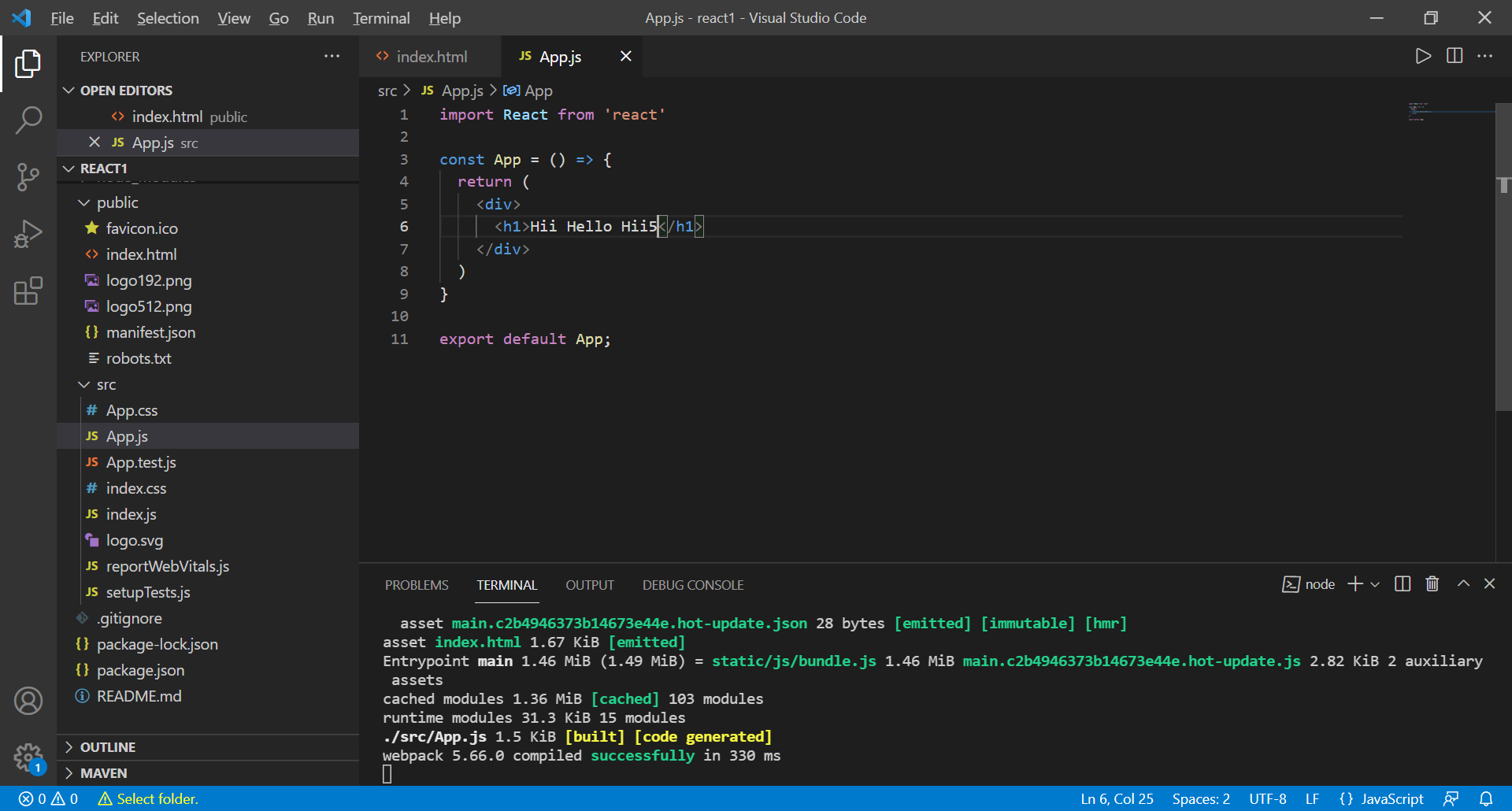
\*package.json -🡪contains details of all packages,modules,and versions are resent here

Below snap shows that, Modification made i.e.,. title is changed and it is reflected in browser.





**DISPLAYING Hii Hello Hii5::**



import React from 'react'

const App = () => {

  return (

    <div>

      <h1>Hii Hello Hii5</h1>

    </div>

  )

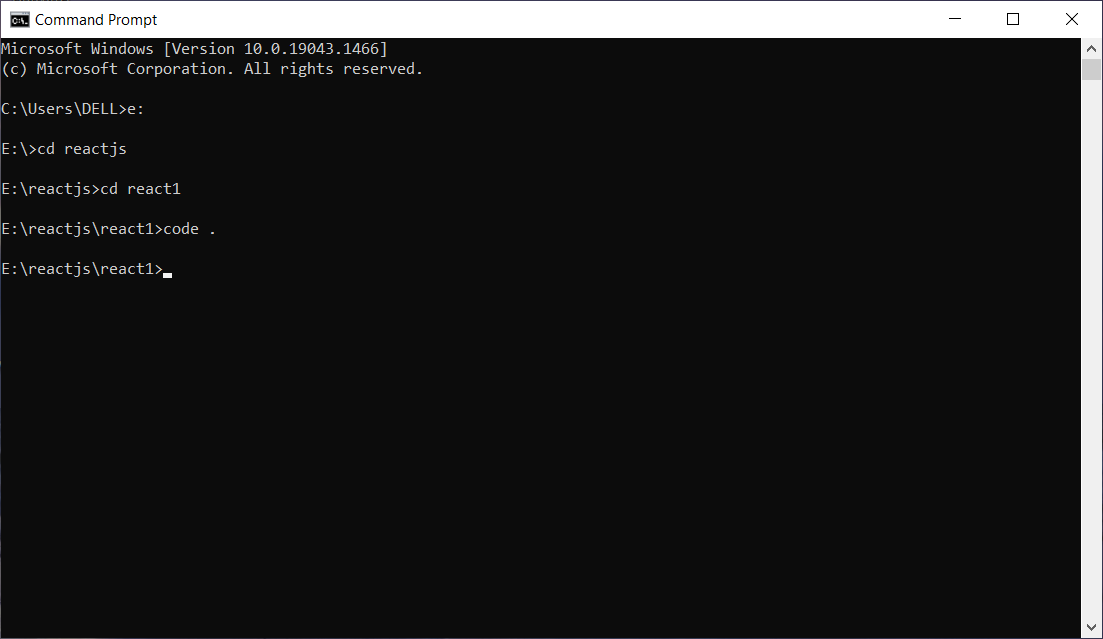
}

export default App;

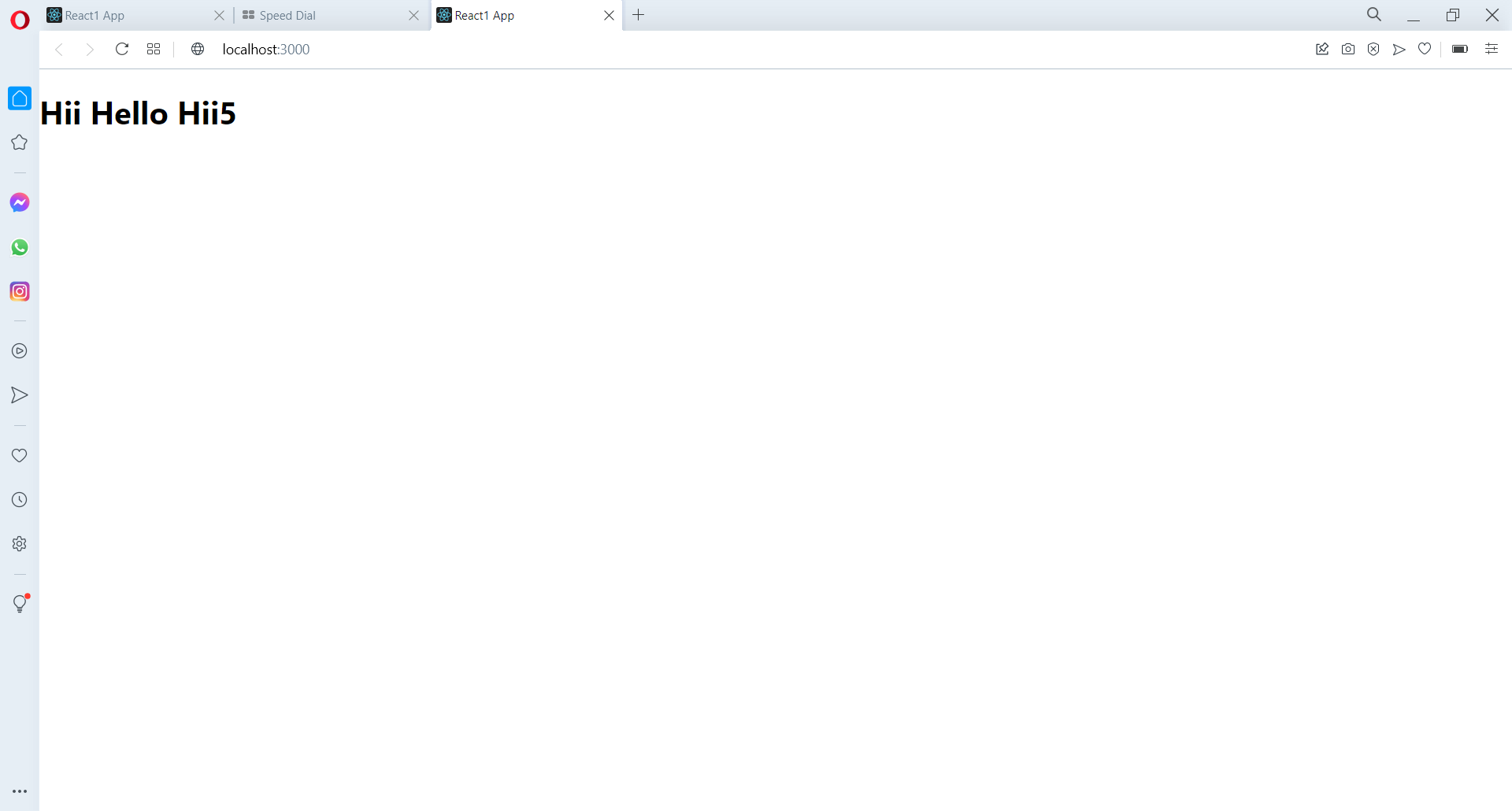
HERE, export default App; this helps to send the App.js data to index.html.



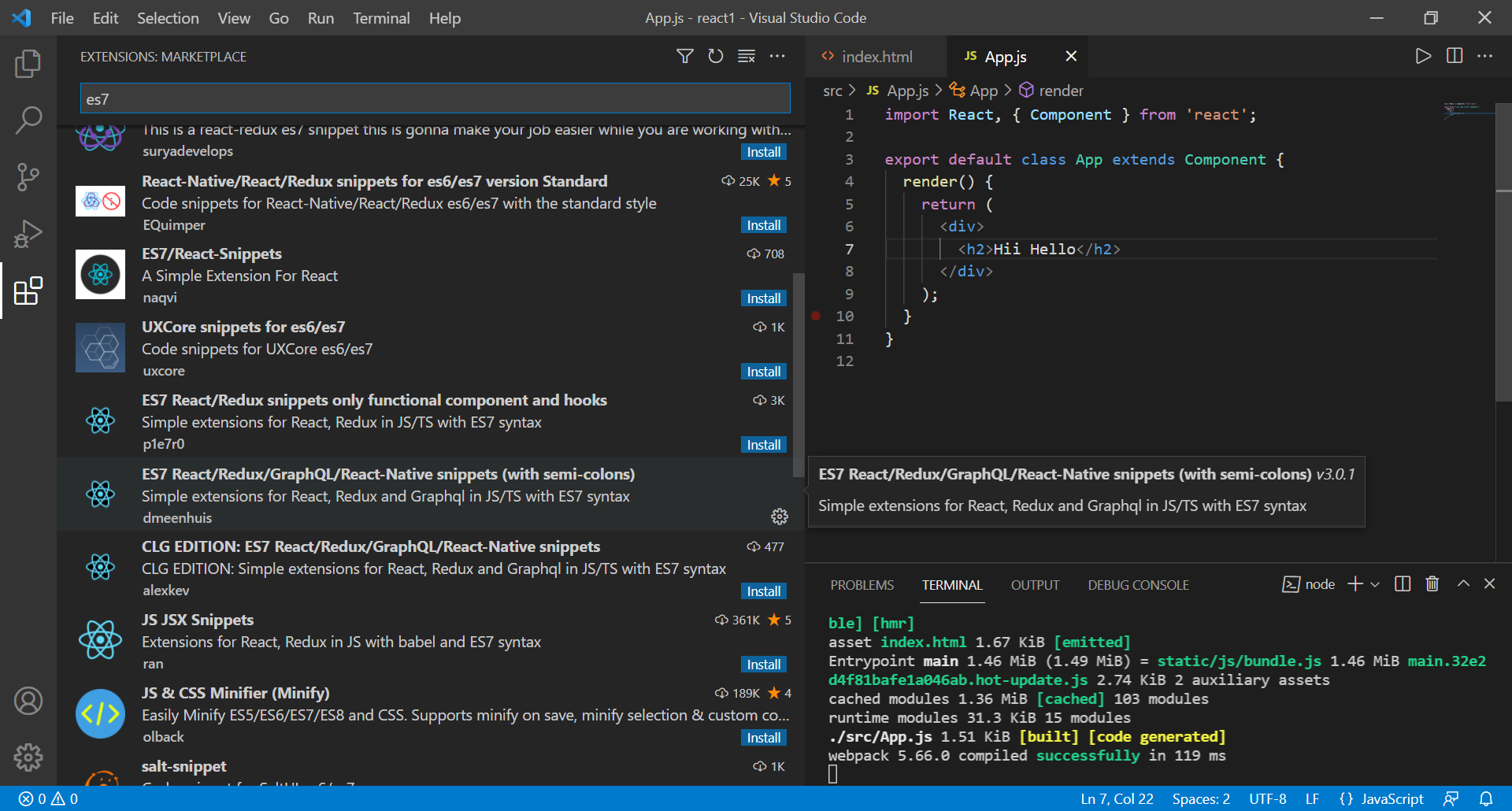
**IF WE WANT TO OPEN ALREADY CREATED PROJECT::**







**EXTENSIONS ARE TO BE INSTALLED IN VISUAL FOR SHORTCUT CODES:**



In App.js :

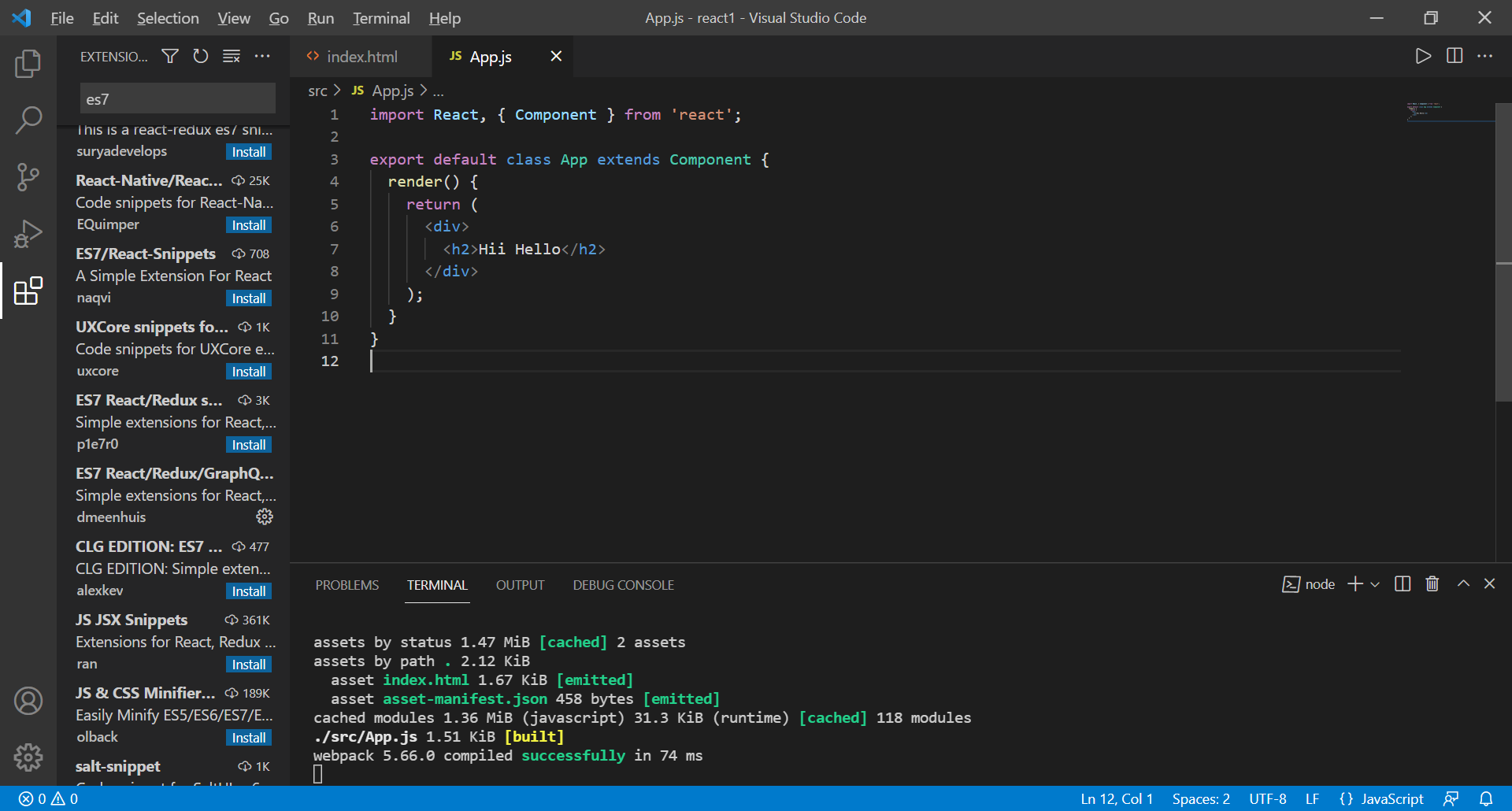
Type -> rafc ->for normal display of text.

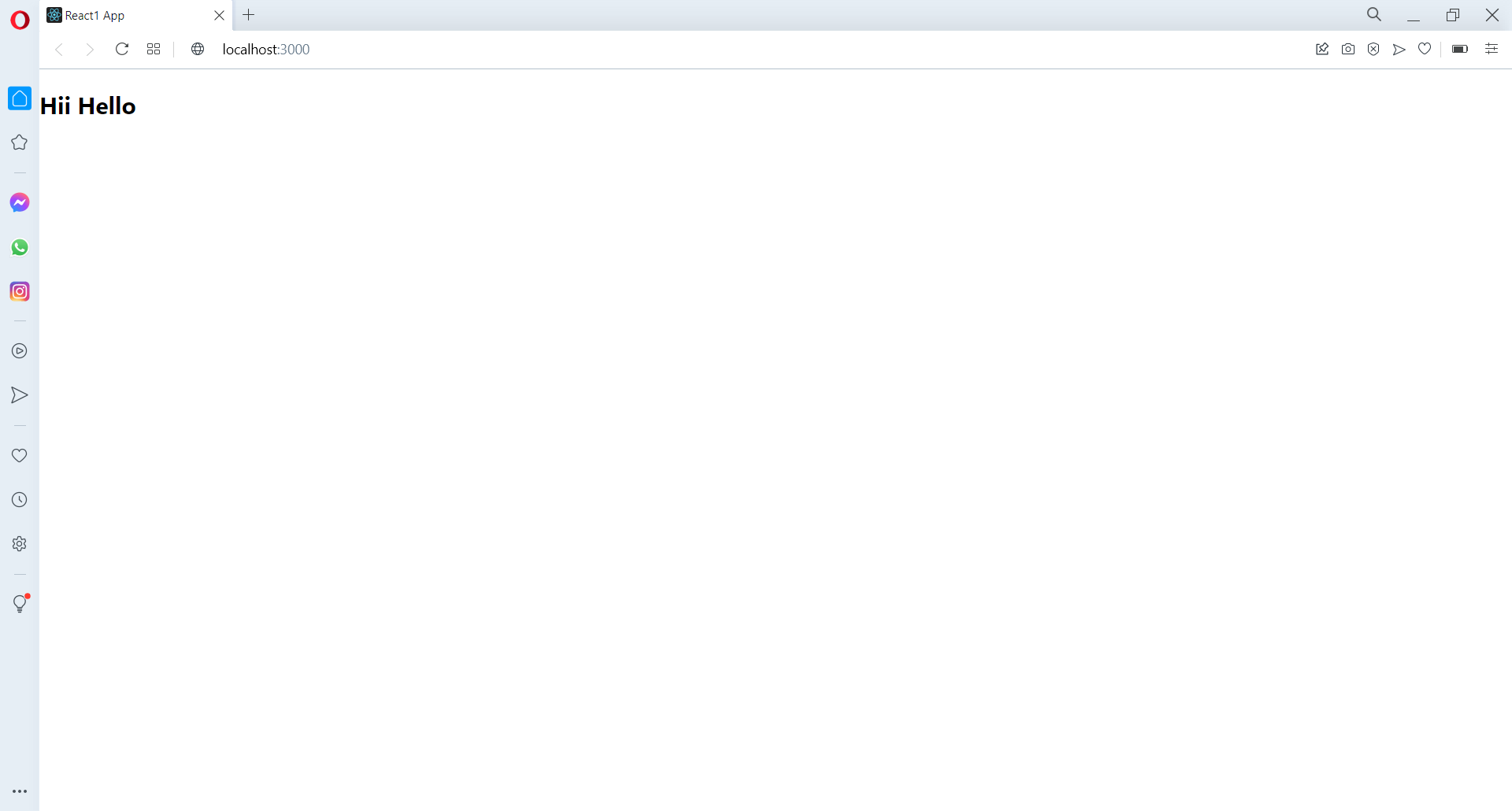
rcc ->for class component ,prop. rsc-> stateless component skeleton

racf ->for function component. rsf->stateless namedfunctionskeleton

**Class component vs Function component::**

Class component: has render(){}





Class component with variable ->(state ={} is used for any variable declaration.)

App.js

import React, { Component } from 'react';

export default class App extends Component {

  state={

    name : "Chandrika",

  }

  render() {

    return (

      <div>

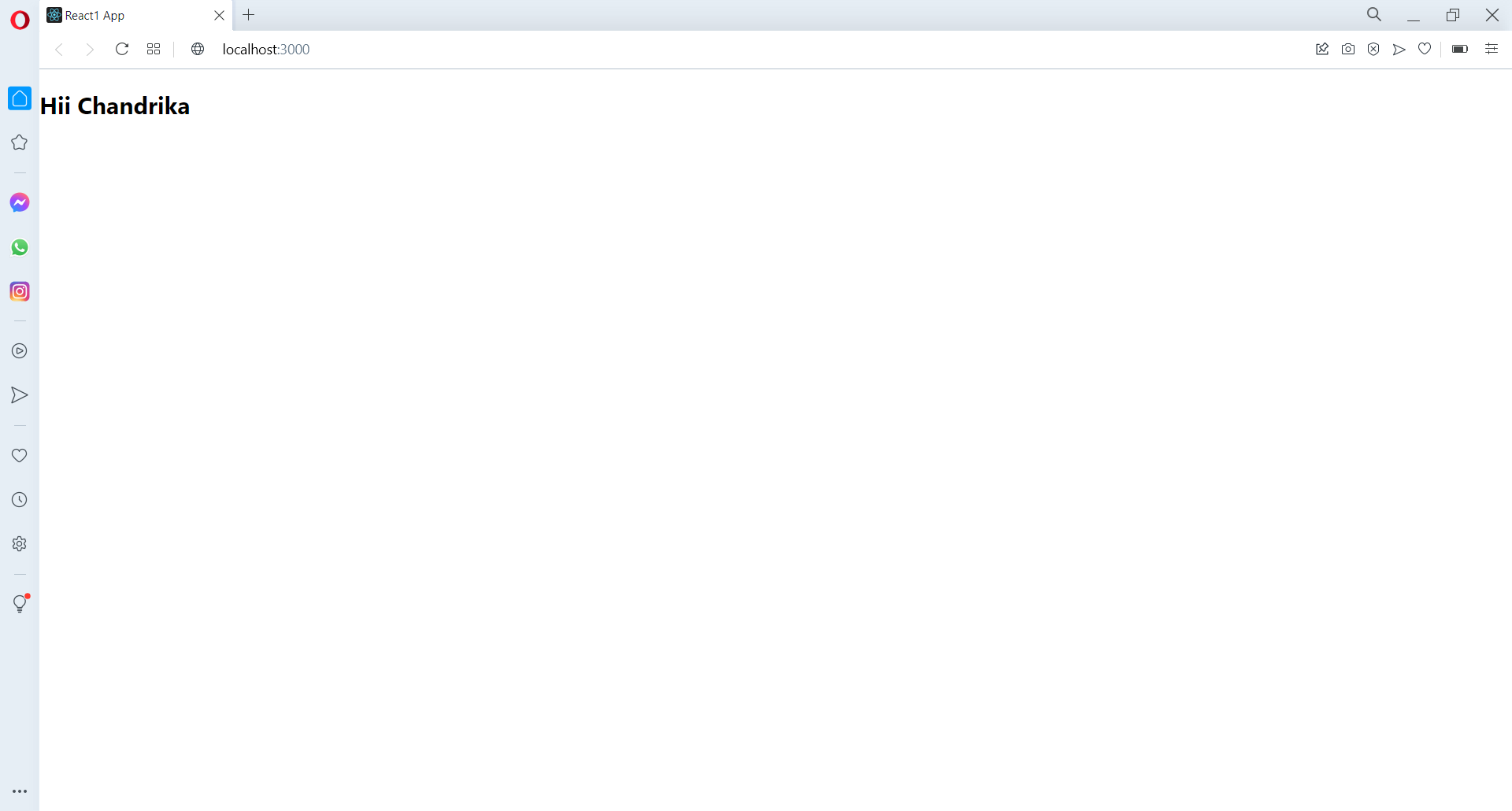
        <h2>Hii {this.state.name}</h2>

      </div>

    );

  }

}



Function component with variable -> has only return and array function i.e.,.const…..

Function component with variable is achieved with the concept of “react hock”

App.js:

import React, {useState} from 'react';

const App = () => {

  const [name,setName] = useState("Marrapu");

  return (

    <div>

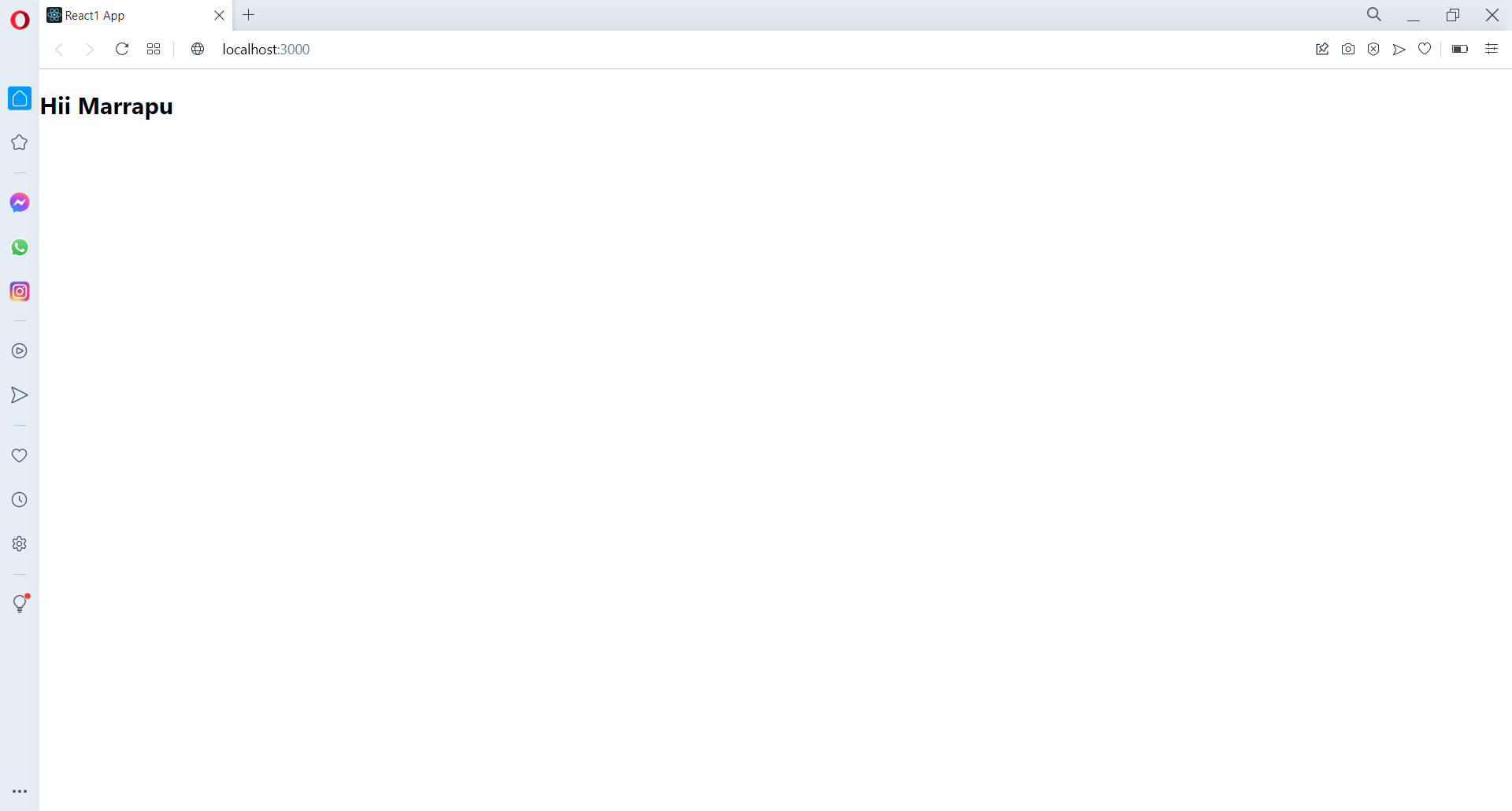
        <h2>Hii {name}</h2>

    </div>

  );

}

export default App;



**State and Prop:**

State ->when we want to declare multiple variable values.

Eg: state is same as class component.

Props ->When we want to send values from one component to another component.

Eg -- 1: for displaying content.

Display.js file ->1st component create in src.

App.js -> 2nd component.

App.js

import React, { Component } from 'react';

import Display from './Display'

export default class App extends Component {

  render() {

    return (

      <div>

        <center>

          <Display />

        </center>

      </div>

    );

  }

}

Display.js

import React, { Component } from 'react';

export default class Display extends Component {

  render() {

    return (

      <div>

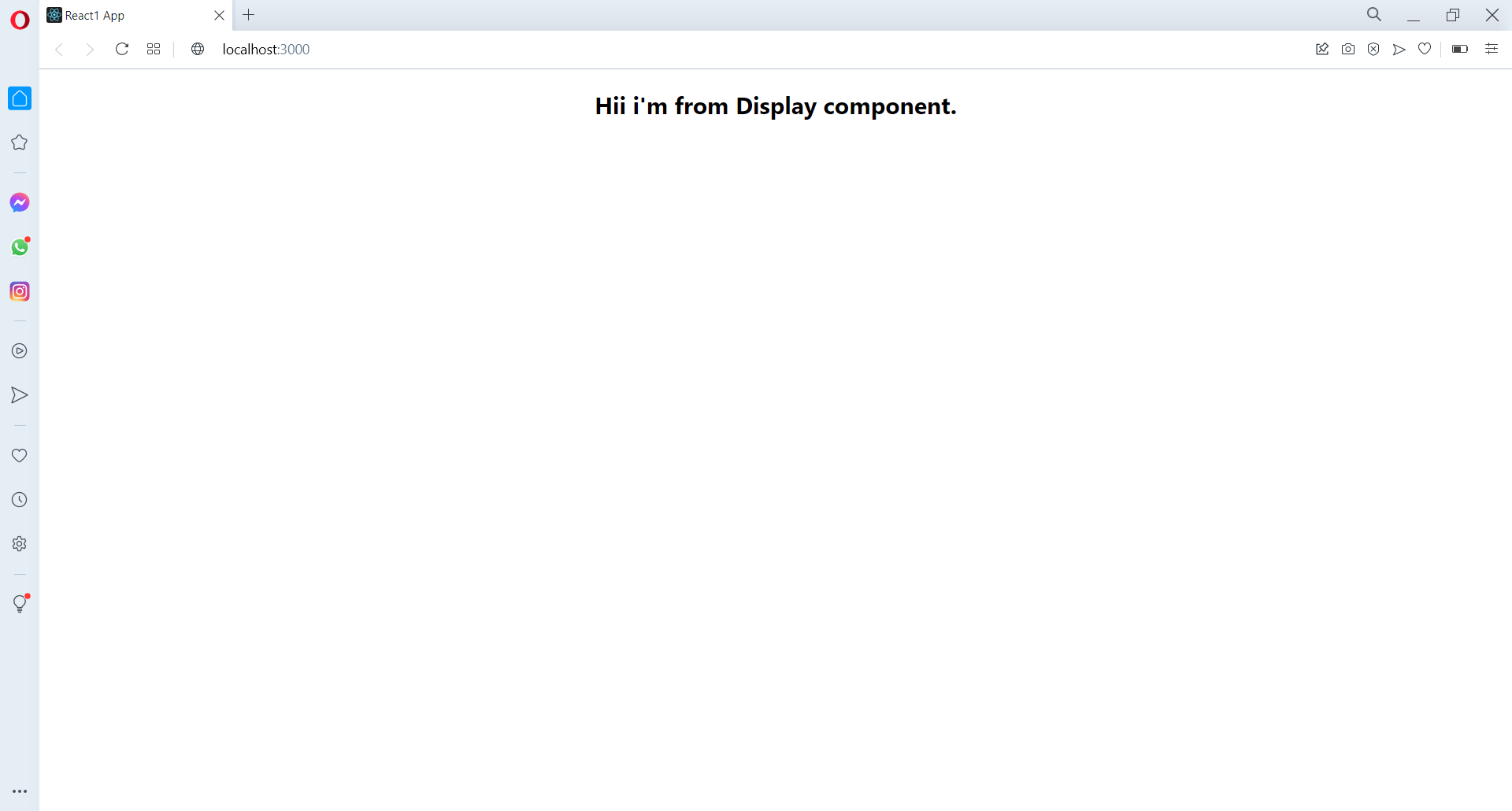
        <h2>Hii i'm from Display component.</h2>

      </div>

    );

  }

}



Eg – 2 : for sending values to Display component from App.js

App.js (parent) -> Display.js (child)

[Variable name] = {this.state.[variable name]} -> {this.prop.[variable name]

App.js:

import React, { Component } from 'react';

import Display from './Display'

export default class App extends Component {

  state = {

    name : "Chandrika"

  }

  render() {

    return (

      <div>

        <center>

          <Display name={this.state.name}/>

        </center>

      </div>

    );

  }

}

Display.js:

import React, { Component } from 'react';

export default class Display extends Component {

  render() {

    return (

      <div>

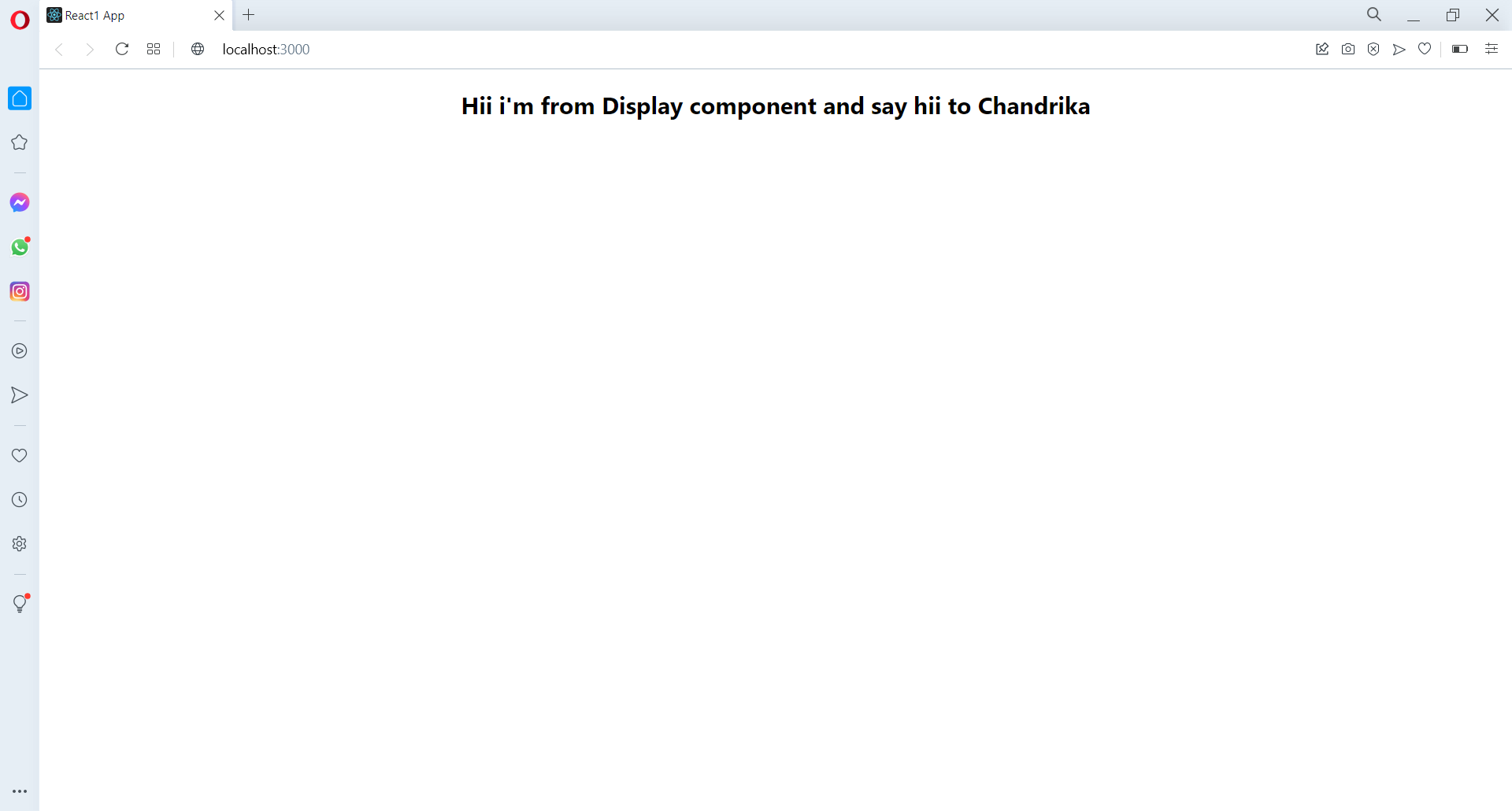
        <h2>Hii i'm from Display component and say hii to {this.props.name}</h2>

      </div>

    );

  }

}



**CSS Styling : 2 ways**

a)Inline style

b)CSS file -- import to -----🡪Main component

HTML + JS = {}

HTML + CSS ={{}}

In div -> a) . –>class b) # -> id

eg for b):

App.js

import React from 'react';

import './App.css';

const App = () => {

  return (

    <div className="container">

      <h1>Hii Chandrika</h1>

    </div>

  );

}

export default App;

App.css

h1{

  color : white;

  background-color: blueviolet;

  padding: 20px;

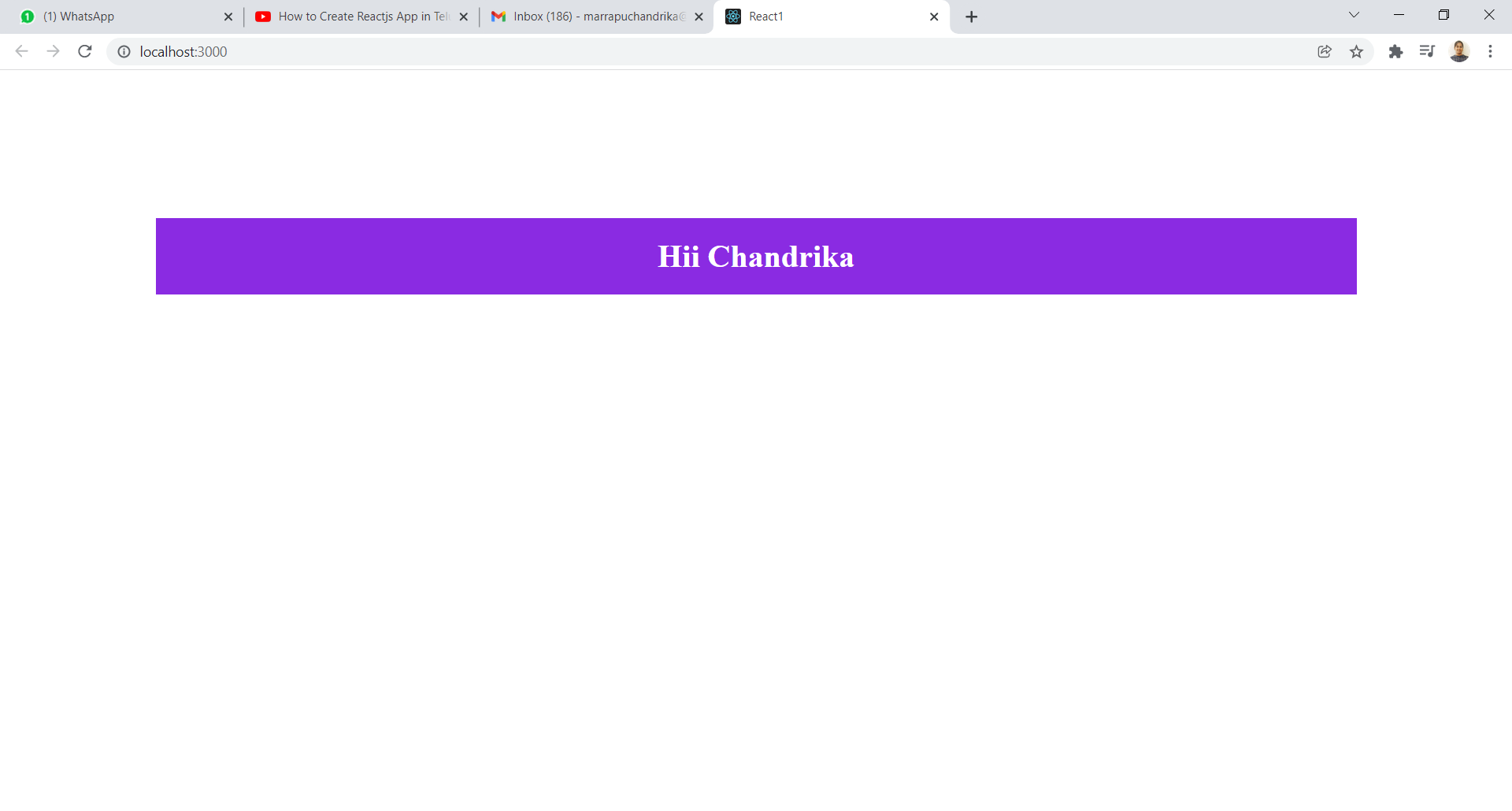
  text-align: center;

}

.container{

  margin: 150px;

}



Eg for a):

App.js

import React from 'react';

const App = () => {

  const styling ={

    color: "White",

    backgroundColor: "red",

    padding: "20px",

    textAlign: "center"

  }

  return (

    <div style={{margin: "100px"}}>

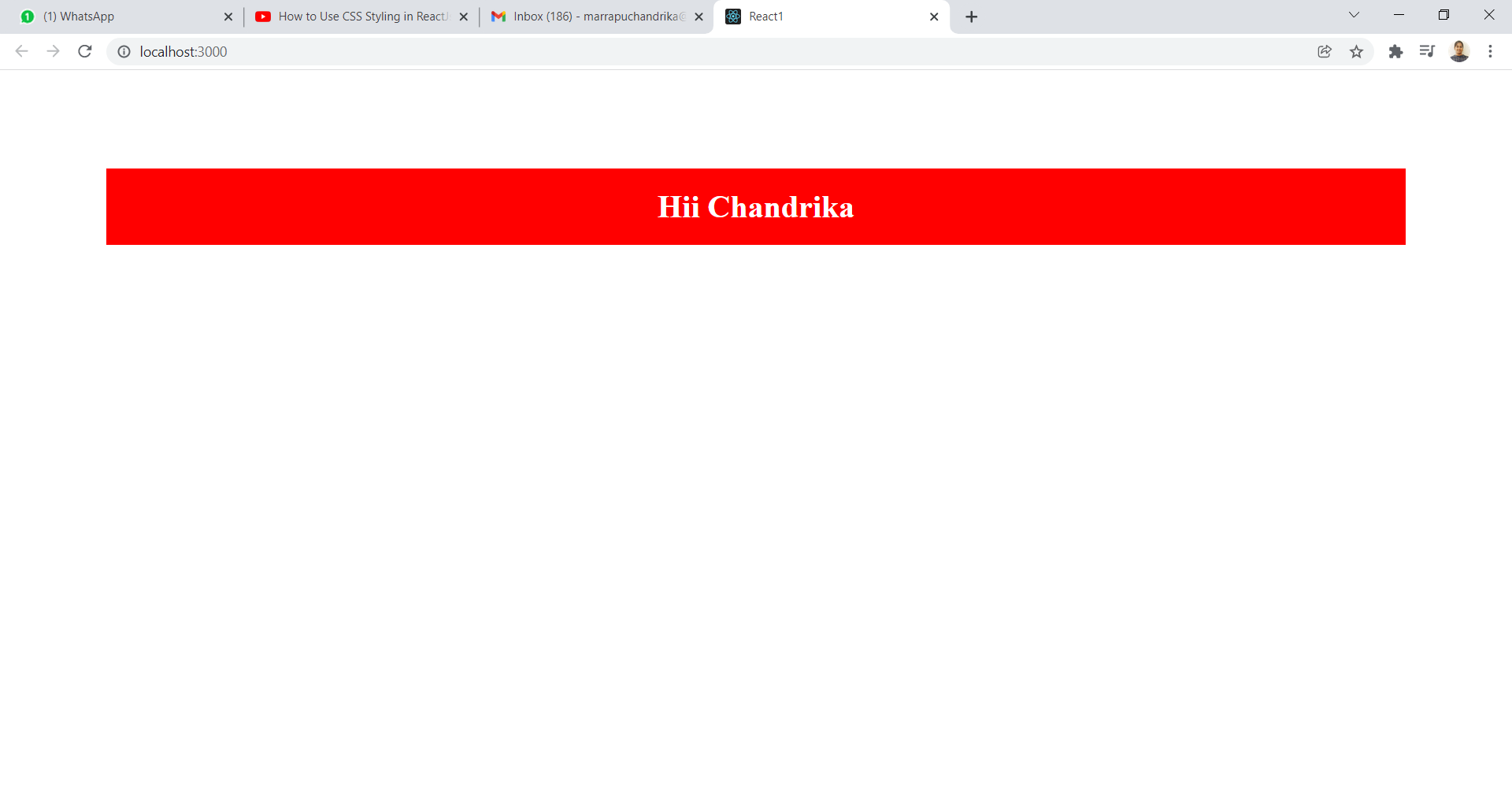
      <h1 style={styling}>Hii Chandrika</h1>

    </div>

  );

}

export default App;



**Onclick event handler : [Button click or Text click]:**

App.js: [button click]

import React from 'react';

const App = () => {

  return (

    <div>

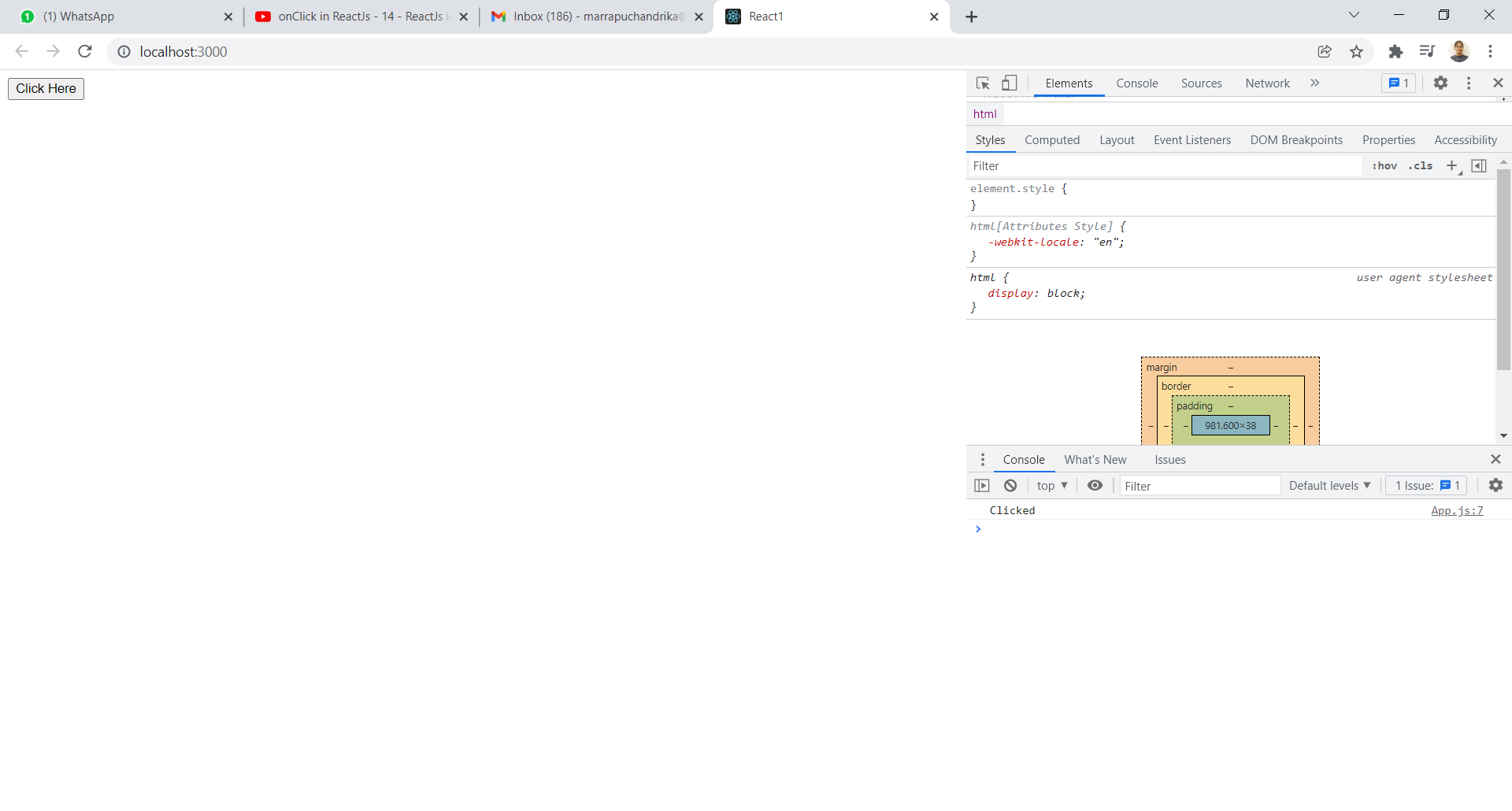
      <button onClick={() =>console.log("Clicked")}>Click Here</button>

    </div>

  );

}

export default App;



App.js :[Text click]

import React from 'react';

const App = () => {

  return (

    <div>

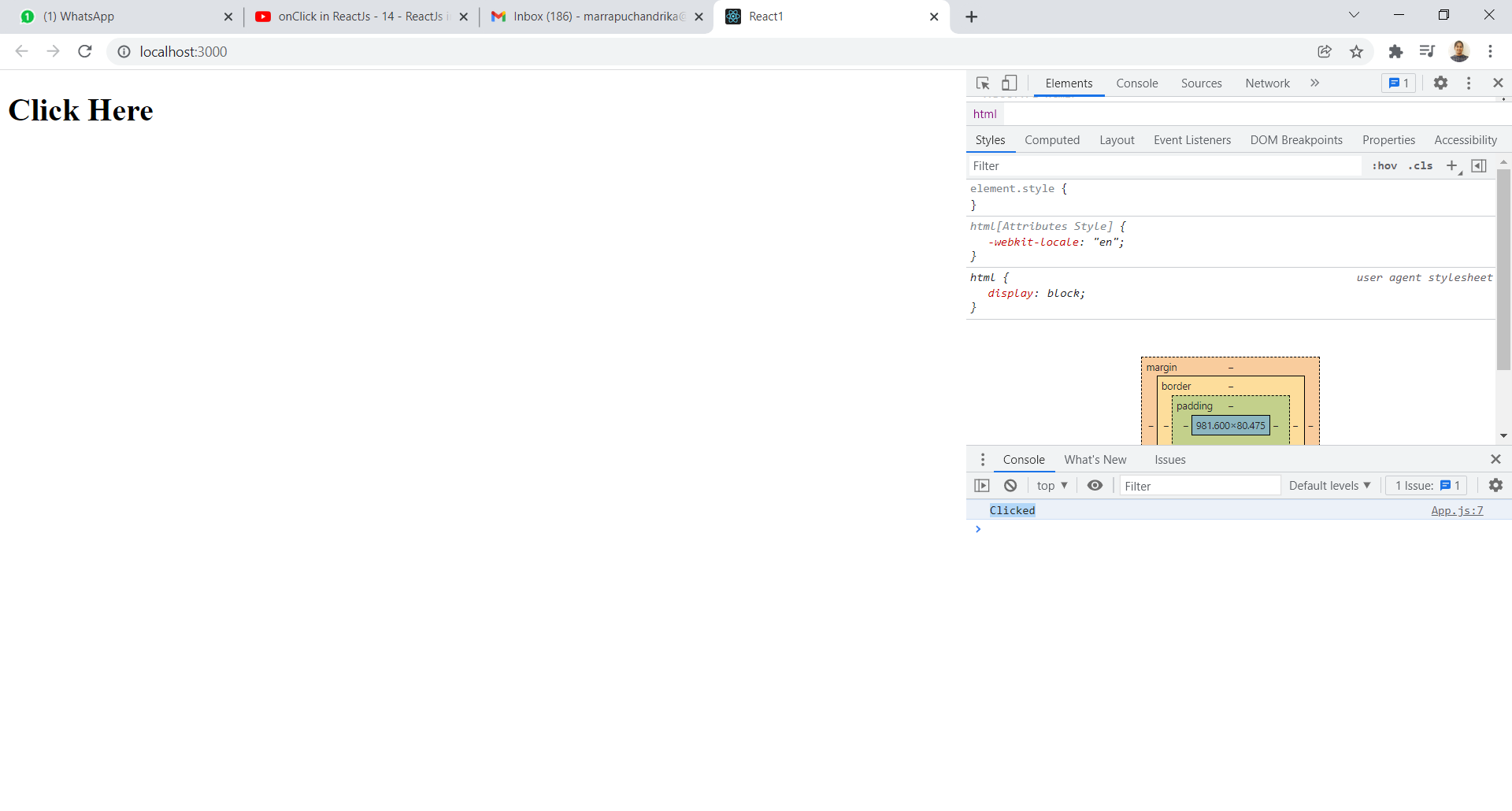
      <h1 onClick={() =>console.log("Clicked")}>Click Here</h1>

    </div>

  );

}

export default App;



**useState in Reactjs:** Used only in Functional level component.

App.js:

import React,{useState} from 'react';

const App = () => {

  const [name,setName] = useState("Know the Marrapu's full name");

  return (

    <div>

      <center>

        <h1>{name}</h1>

          <button onClick={() => setName("Marrapu Chandrika")}>What?</button>

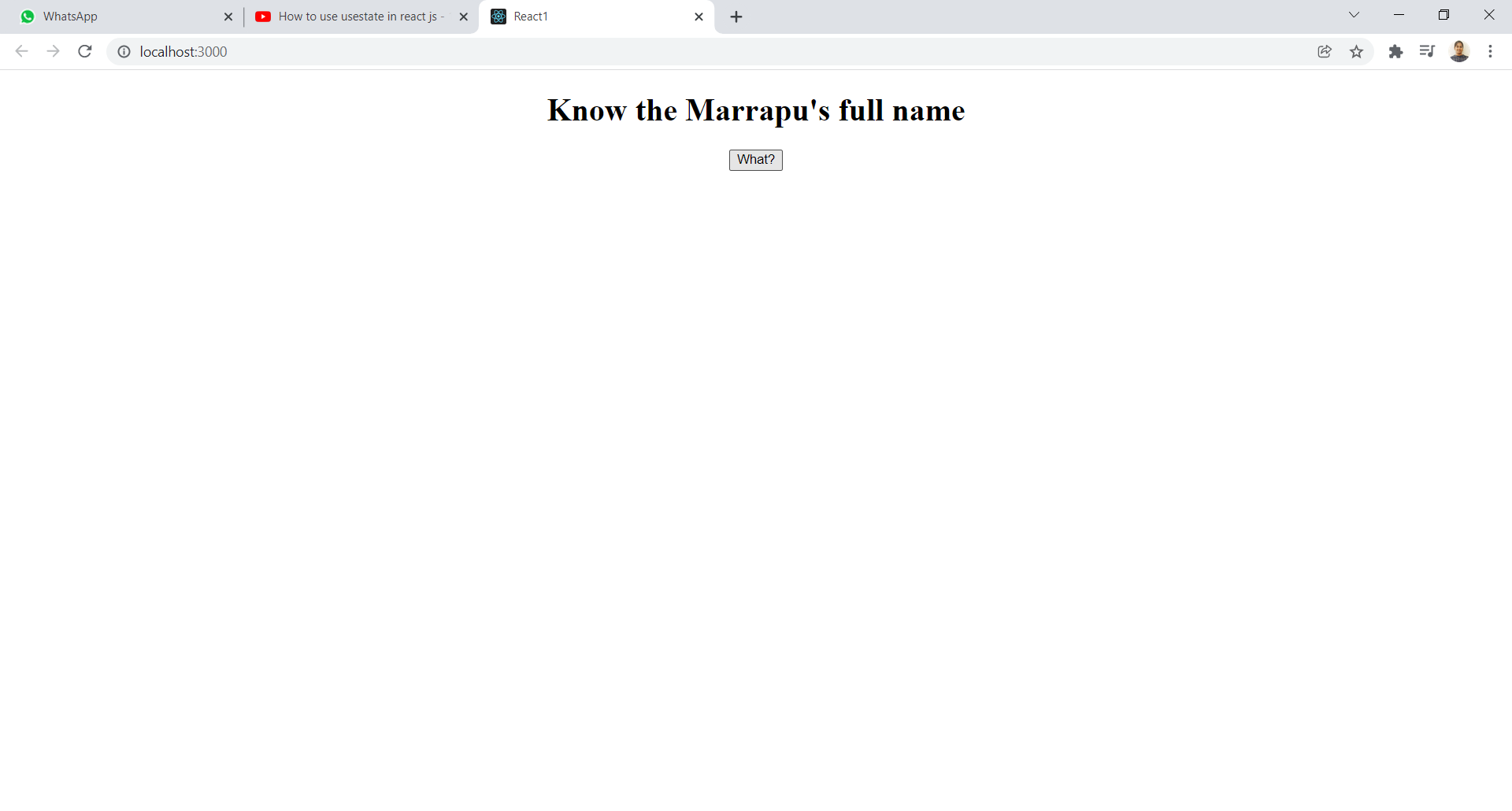
      </center>

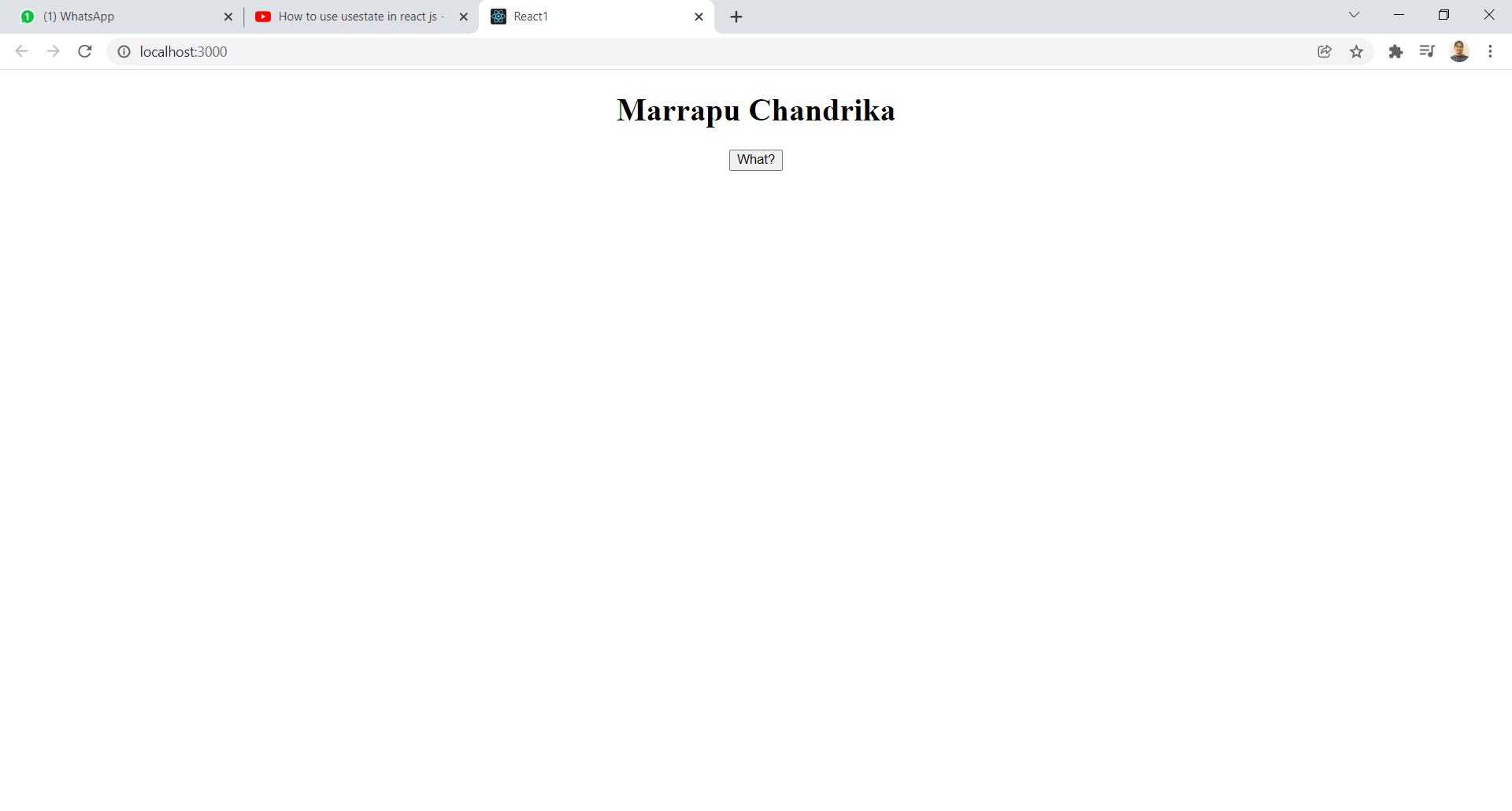
    </div>

  );

}

export default App;





Another eg for useState: if we want to know the button is clicked or not..(incerement).

App.js:

import React,{useState} from 'react';

const App = () => {

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

  return (

    <div>

      <center>

        <h1>{count}</h1>

          <button onClick={() => setCount(count+1)}>Click</button>

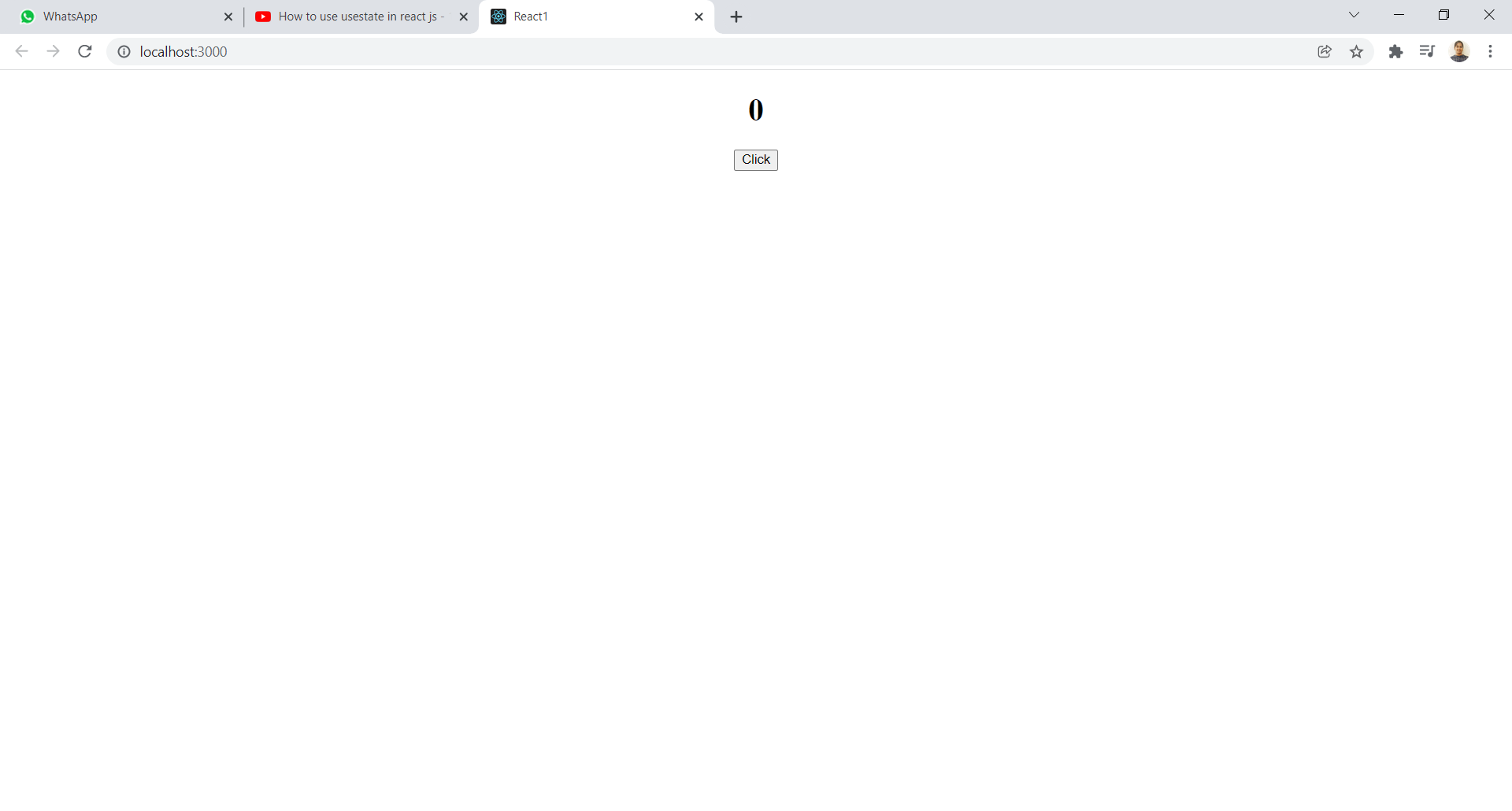
      </center>

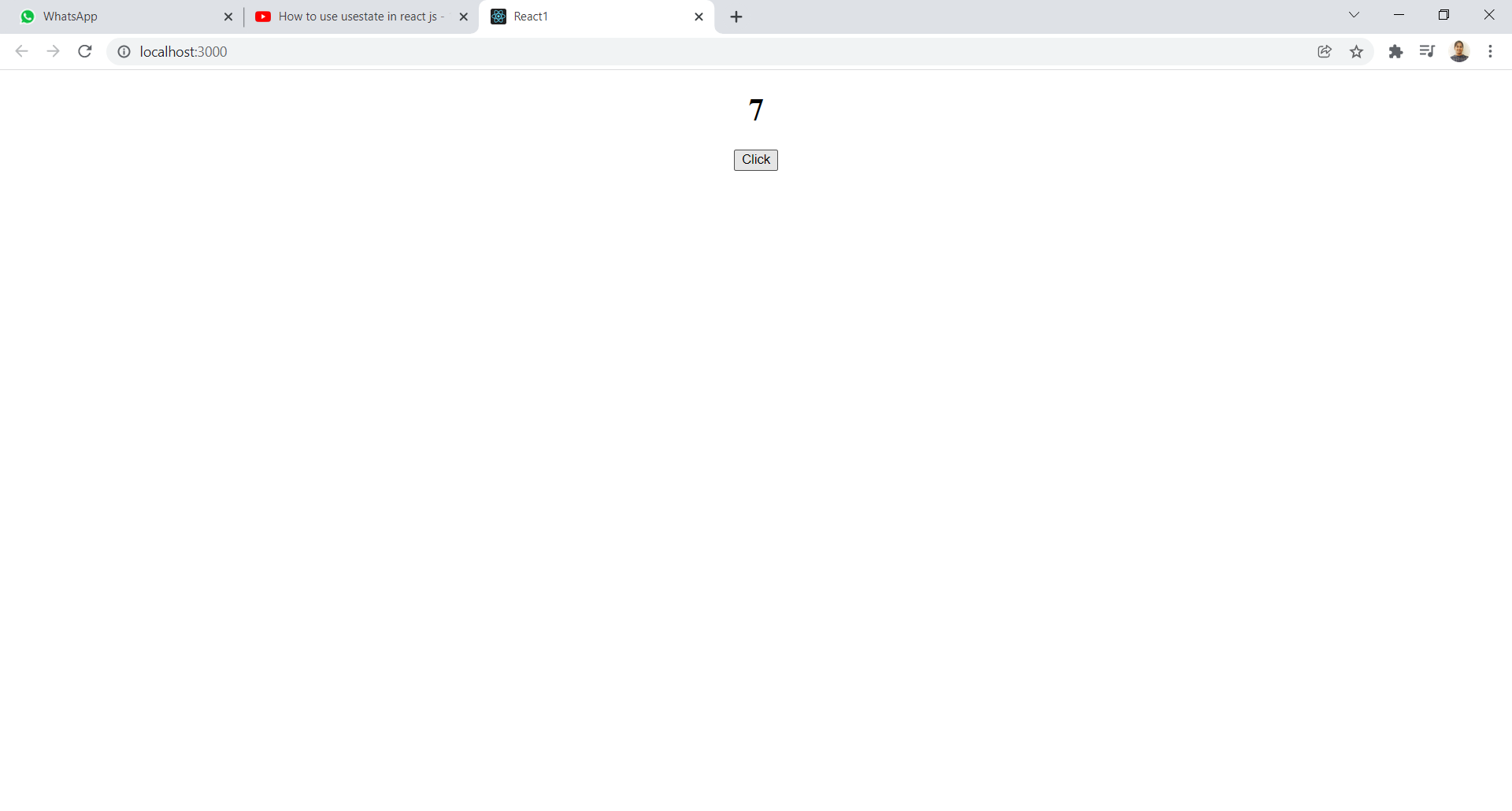
    </div>

  );

}

export default App;





**useEffect:** Real Time hoock-Function Component- It is used when we want to execute any function after return statement.

App.js:

Here,useEffect is executed after return statement is printed.It takes two parameters a)function b)dependencies- it means like incerement.

import React,{useState,useEffect} from 'react';

const App = () => {

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

  useEffect(() => console.log("clicked"),[])

  return (

    <div>

      <center>

        <h1>{count}</h1>

          <button onClick={() => setCount(count+1)}>Click Me</button>

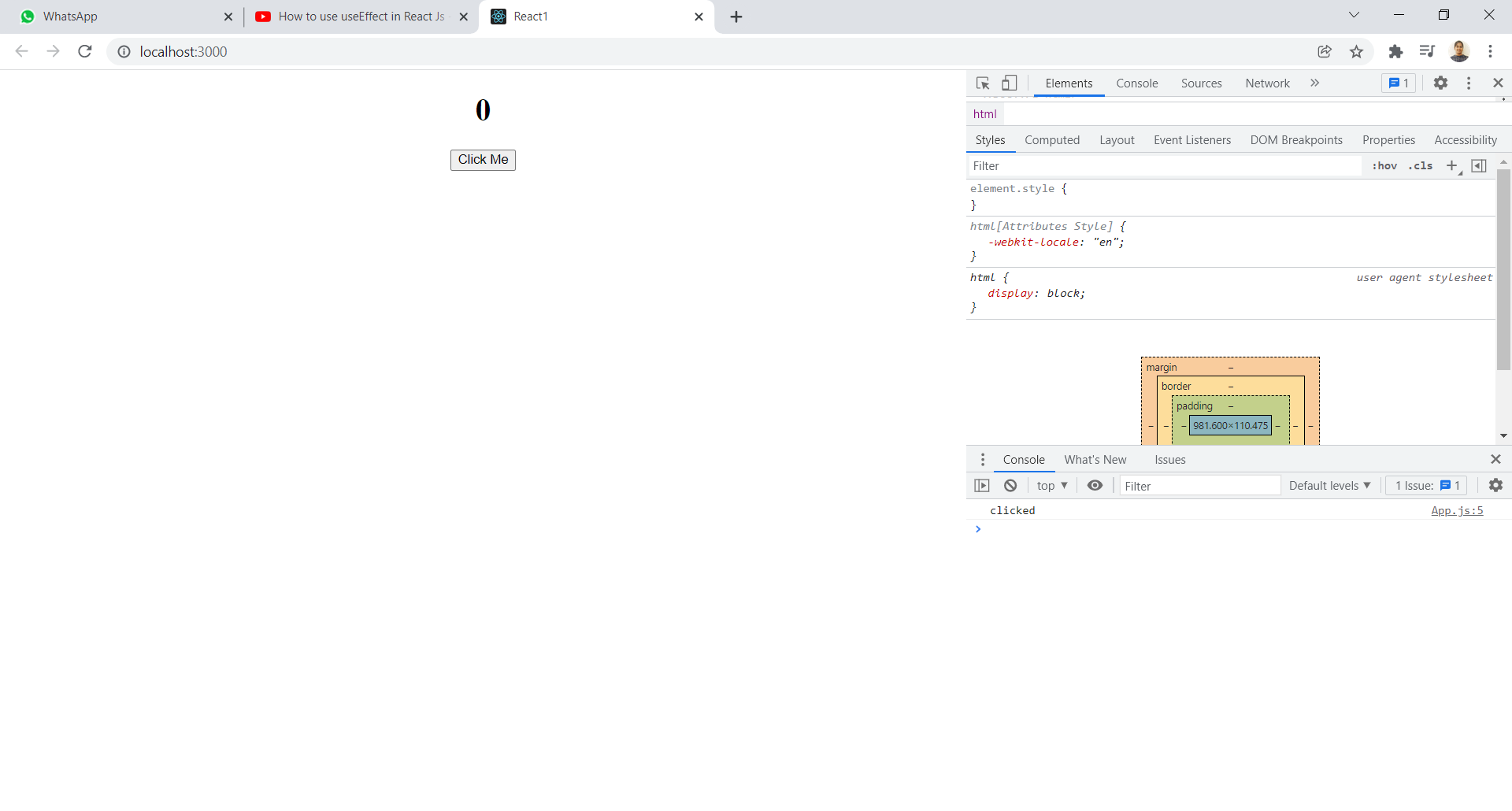
      </center>

    </div>

  );

}

export default App;



**Eg2:** used dependency as count-whenever onclick event executed those many times useEffect gets printed.

import React,{useState,useEffect} from 'react';

const App = () => {

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

  useEffect(() => console.log("clicked"),[count])

  return (

    <div>

      <center>

        <p>you clicked {count} times</p>

          <button onClick={() => setCount(count+1)}>Click Me</button>

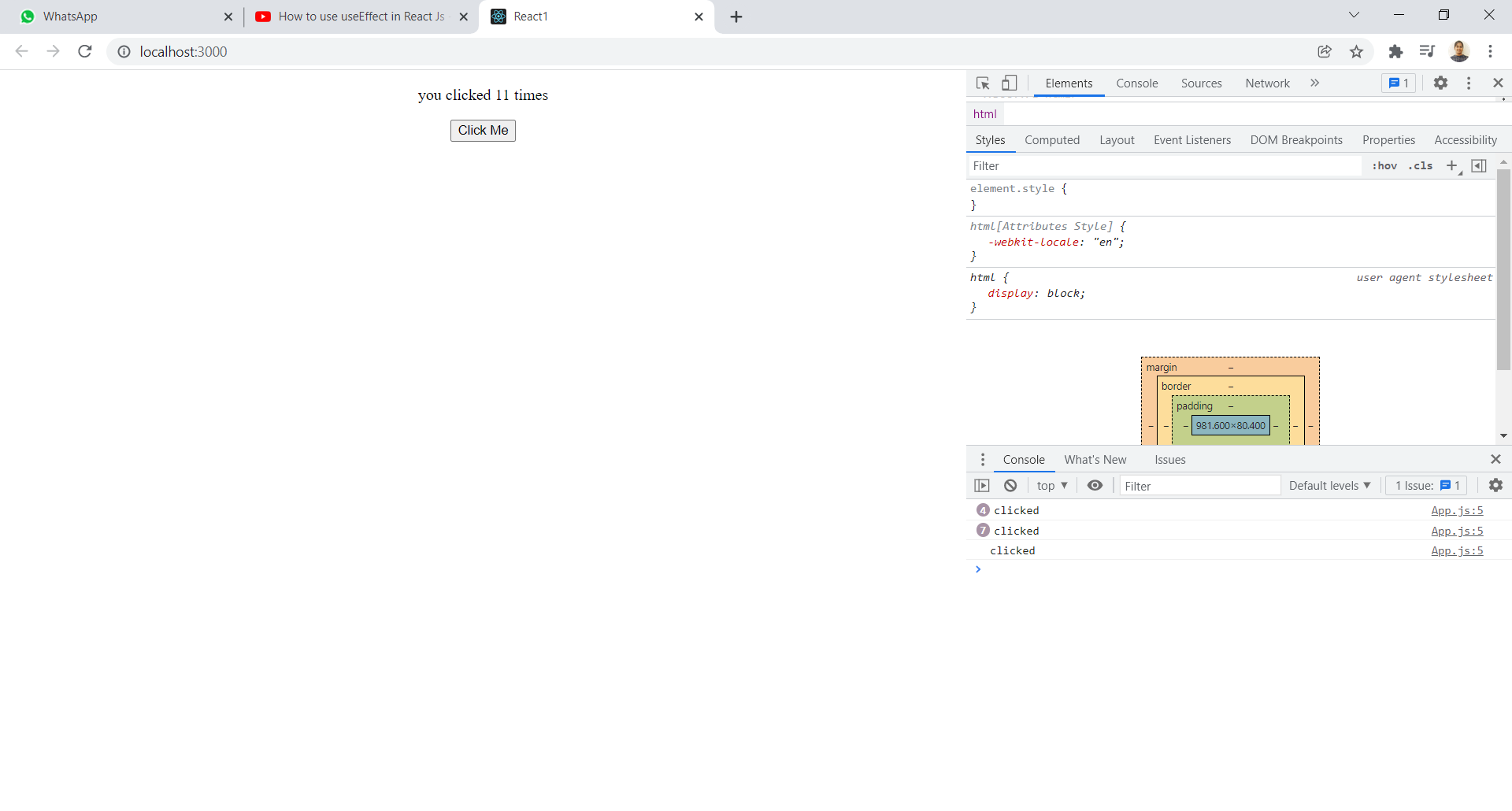
      </center>

    </div>

  );

}

export default App;



Eg3:Here function and dependency is given as count.

import React,{useState,useEffect} from 'react';

const App = () => {

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

  useEffect(() => console.log(count),[count])

  return (

    <div>

      <center>

        <p>you clicked {count} times</p>

          <button onClick={() => setCount(count+1)}>Click Me</button>

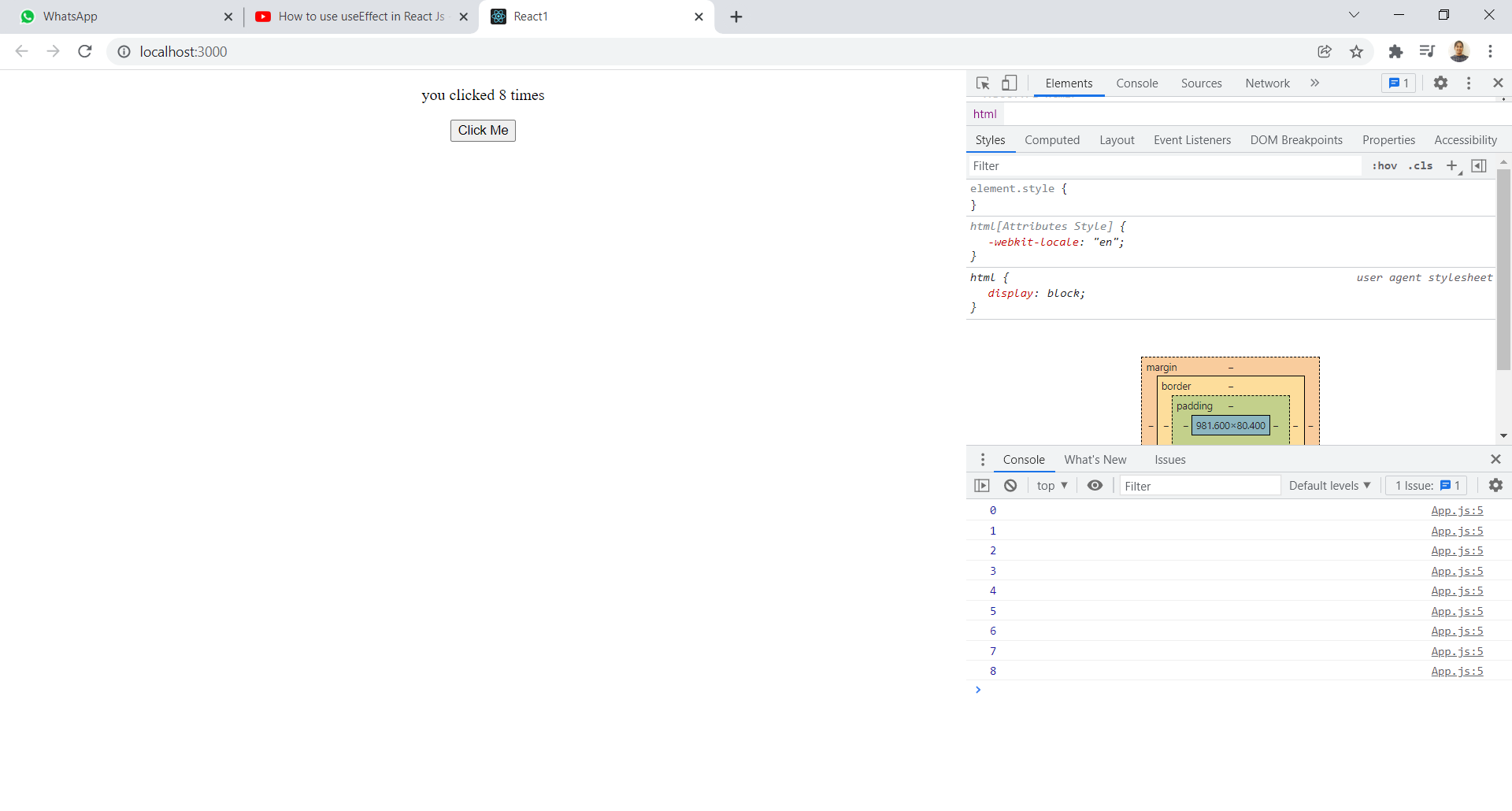
      </center>

    </div>

  );

}

export default App;



**onChange event handler:** Used to handle and assign the user input data to a variable.

Here,

1. const [user,setUser] = useState("");

it is used to store the user input to user variable.

2)we use onChange event handler in input tag and assigns input to user variable by value => value={user} by using setUser.

import React,{useState} from 'react';

const App = () => {

  const [user,setUser] = useState("");

  const handler = e =>{

    setUser(e.target.value)

  }

  return (

    <div>

      <center>

        <input type="text" placeholder="Enter Username" value={user} name="user" onChange={handler}/>

        <br></br>

        {user}

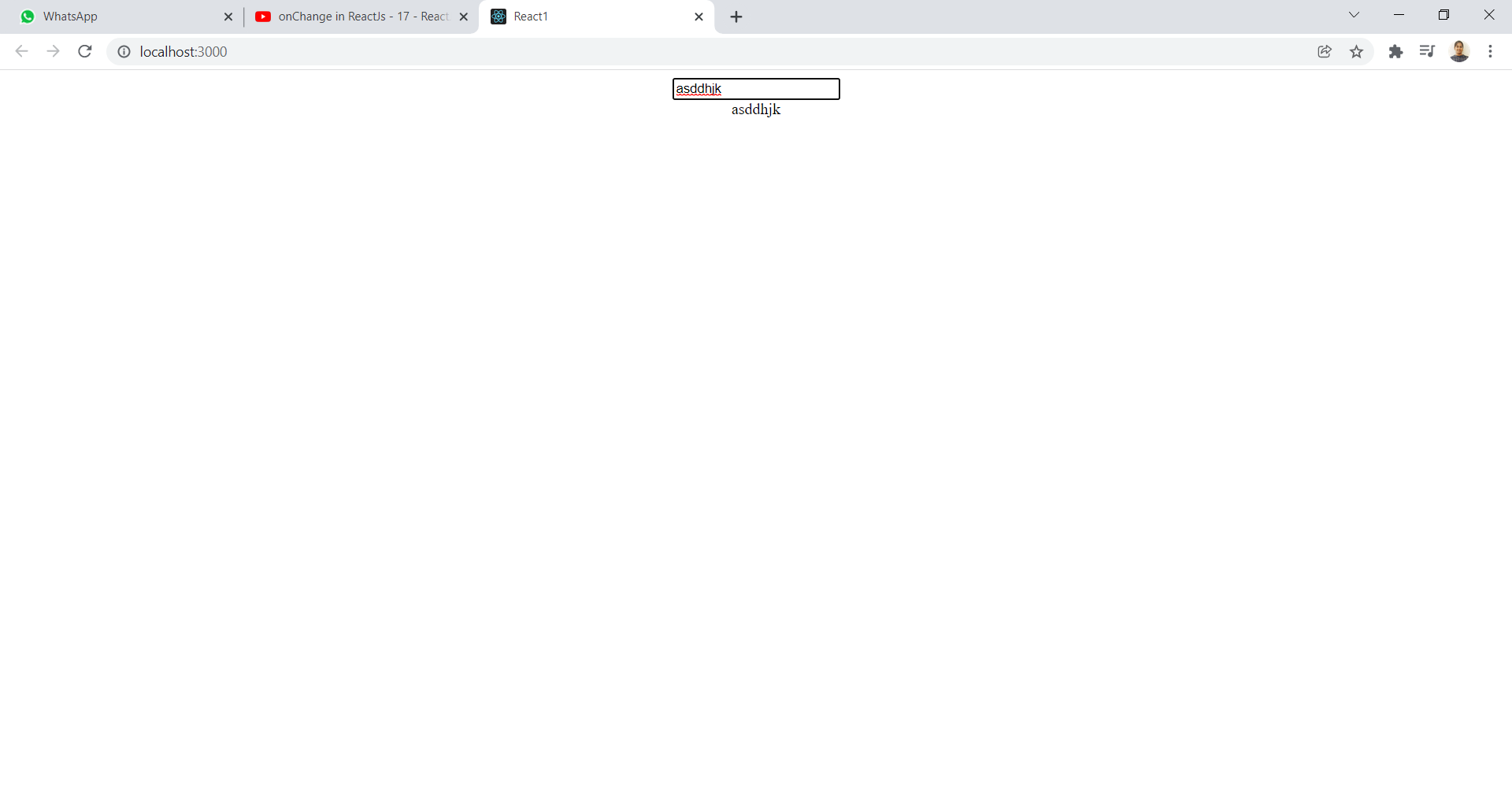
      </center>

    </div>

  );

}

export default App;



**onSubmit Handler:** Used to store the form input data in backend and display.

NEED 3 :

1)useEffect-when we want to execute any function after return statement.

2)Destructuring the data-set of values are unpacked and assign to the individual variables.

3)onChange-used to handle the data and assign to variable.

import React ,{useState} from 'react';

function App () {

  const [data,setData] = useState({

    username : '',

    password : '',

  })

  const{username,password} = data;

  const onChange = e => {

    setData({...data,[e.target.name] : [e.target.value]})

  }

  const submitHandler = e =>{

    e.preventDefault();

    console.log(data);

  }

  return (

    <div>

      <center>

        <form onSubmit={submitHandler}>

          <input type="text" name="username" value={username} onChange={onChange} placeholder='Enter Username'/><br/>

          <input type="password" name="password" value={password} onChange={onChange} placeholder='Enter password'/><br/>

          <input type="submit" name="SUBMIT"/>

        </form>

      </center>

    </div>

  );

}

export default App;

HERE,

1)const [data,setData] = useState({

username : '',

password : '',

})

=>{username and password} are objects in function.

Assigning usrename and password to data(varaible).

2) function App () {

const [data,setData] = useState({

username : '',

password : '',

})

const{username,password} = data;

=>Destructuring the data.

3) const onChange = e => {

setData({...data,[e.target.name] : [e.target.value]})

}

=>onChange event Handler.

4) const submitHandler = e =>{

e.preventDefault();

console.log(data);

}

=>When we want to store data,we use submit handler.

5) e.preventDefault();

=>It stops all the triggered events.

6) <form onSubmit={submitHandler}>



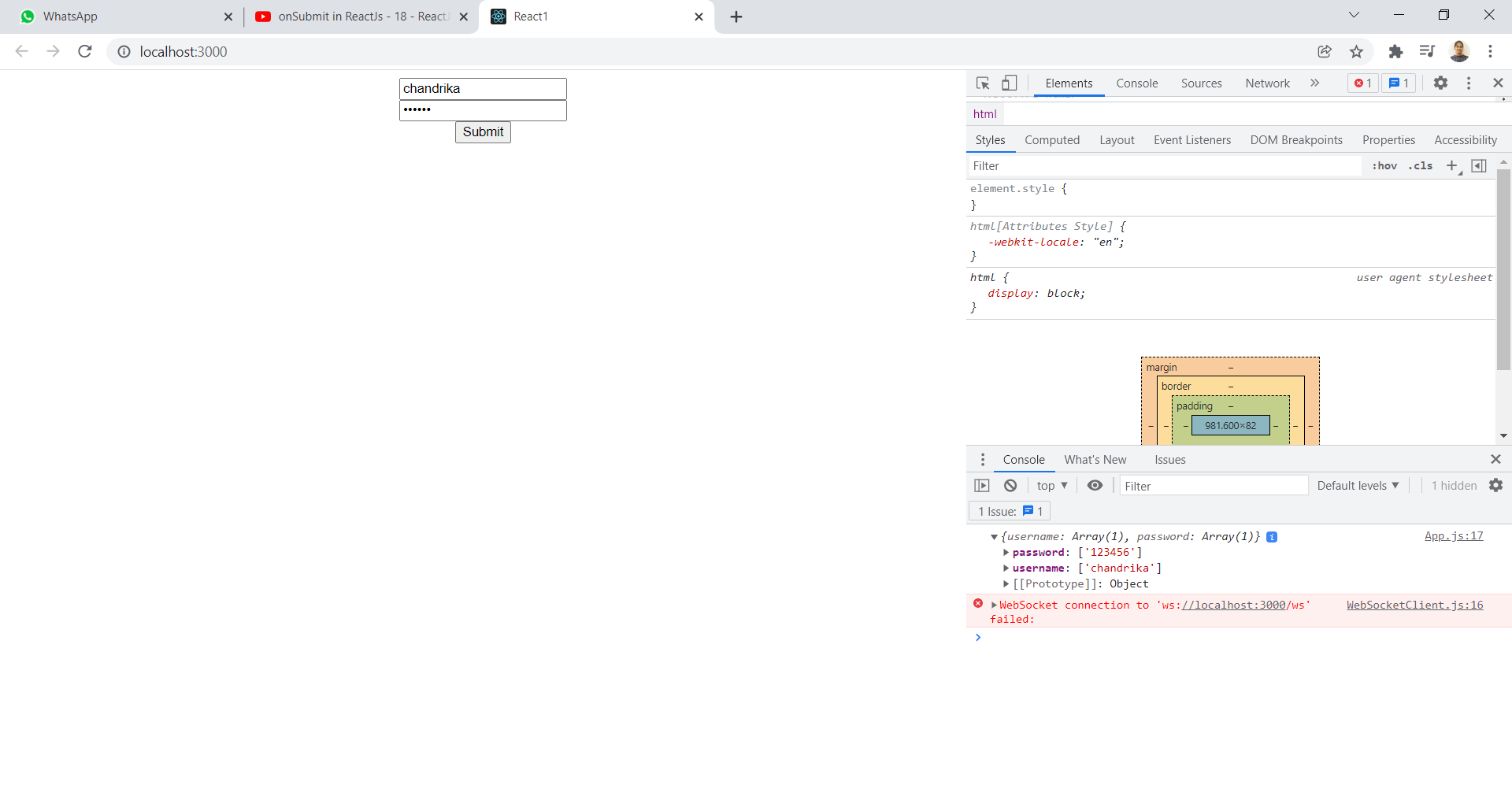
=>when we click on SUBMIT button ,then {submitHandler}works.

Submithandler{function executes}



Data{variable} EXECUTES.





**Map() function:** We can manipulate the items in an array by iterating and accessing individual items.

Syntax: array.map((value,index) => <li> {value} </li>)

Eg 1-

const num = [3, 8, 11, 7, 5];

2

3const num2x = num.map((n) => n \* 2);

4

5console.log(num2x); // [6, 16, 22, 14, 10]

Eg-2:App.js (MODEL-1)

import React from 'react';

function App () {

  const arr = ["React js","Node js","Angular js"]

  return (

    <div align='center'>

      {

        arr.map(

          (value,index) => <li>{value}</li>

        )

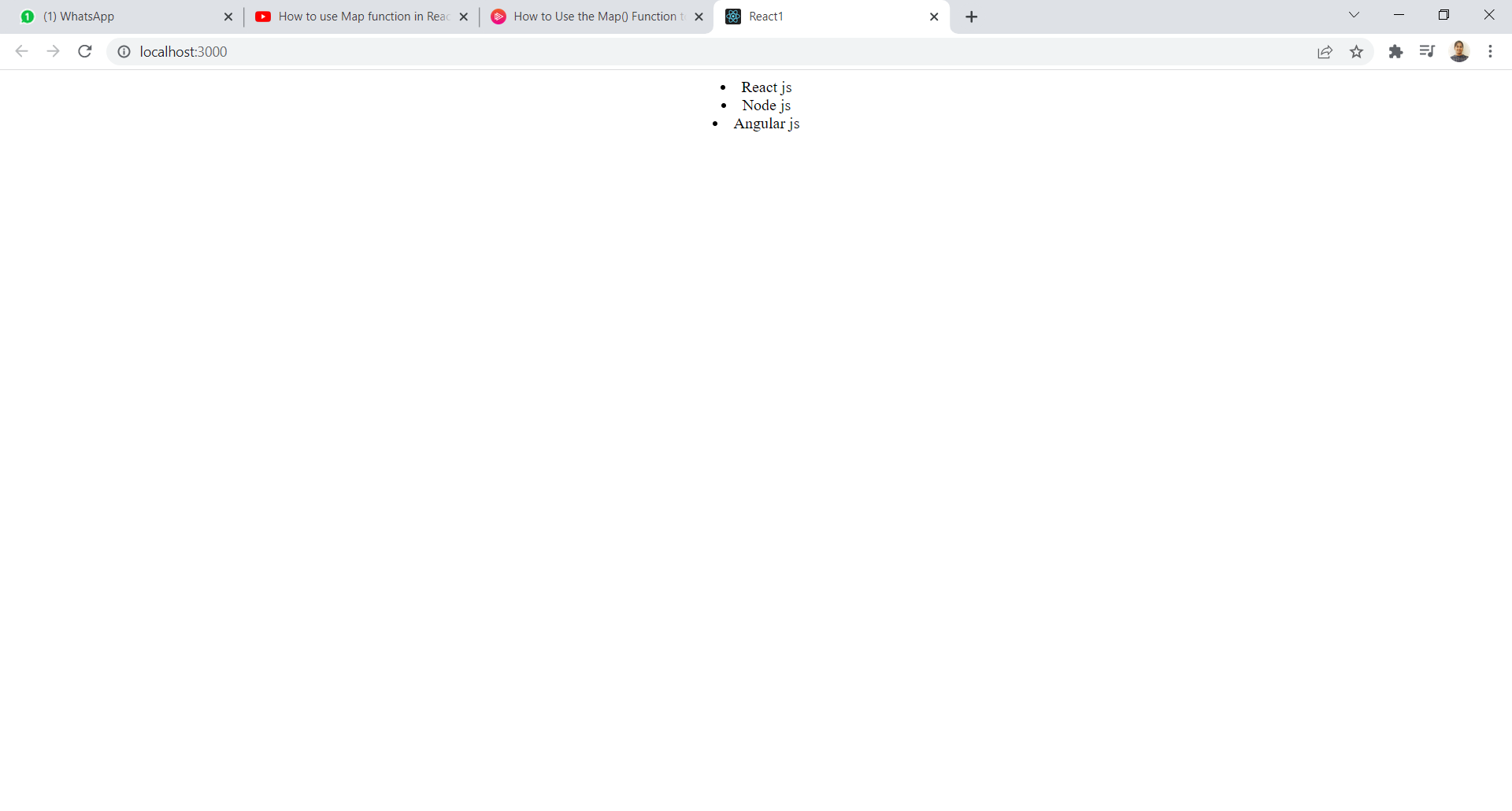
      }

    </div>

  );

}

export default App;



Eg-3: App.js (MODEL-2) -For array of objects

=>4objects, each object has 2 attributes id,title.

import React from 'react';

function App () {

  const arr = [

    {

      id:1,

      title:"Reactjs"

    },

    {

      id:2,

      title:"Nodejs"

    },

    {

      id:3,

      title:"Angularjs"

    },

    {

      id:4,

      title:"Expressjs"

    }

  ]

  return (

    <div align='center'>

      {

        arr.map(

          (value) => <li key={value.id}>{value.title}</li>

        )

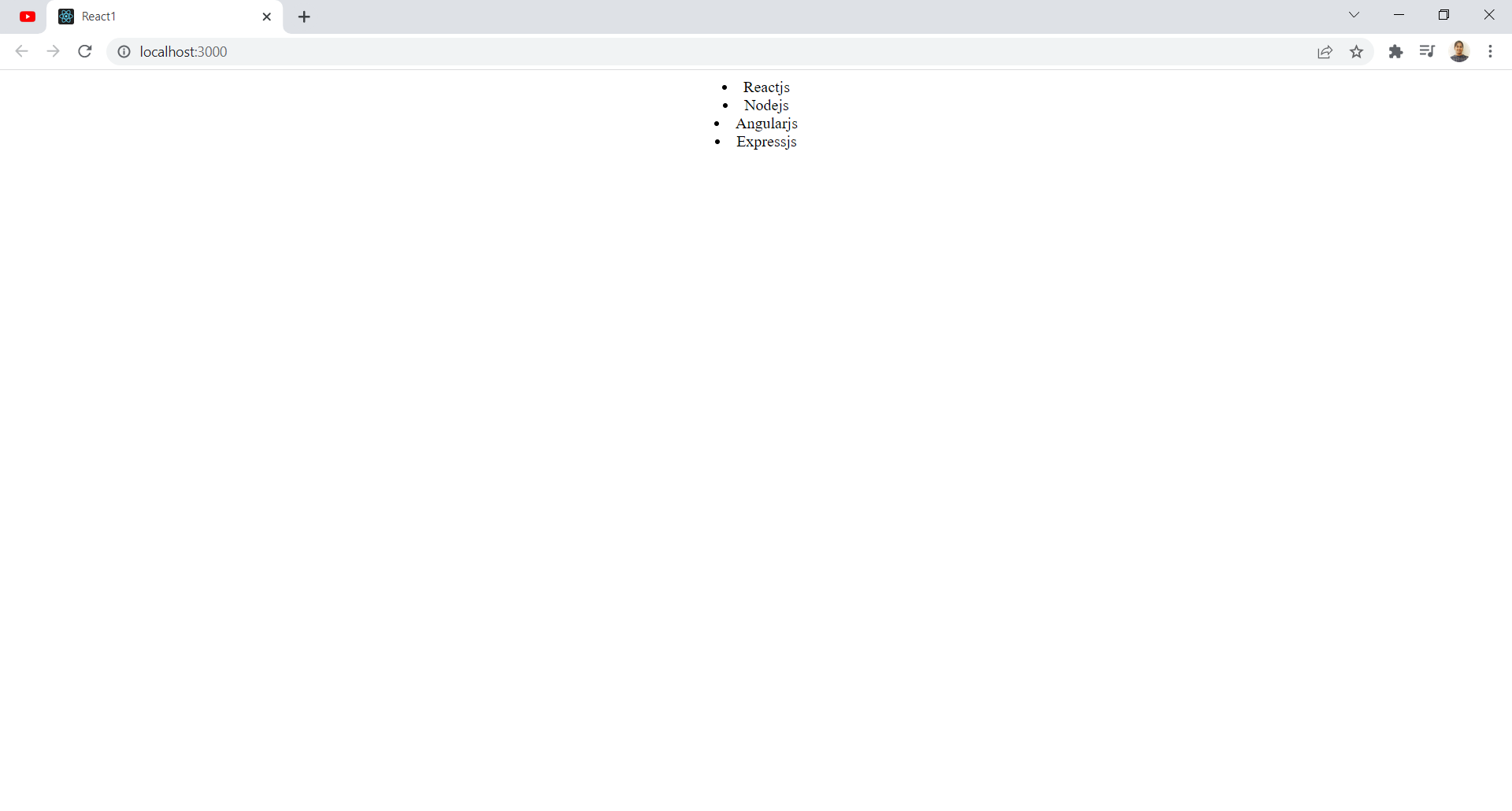
      }

    </div>

  );

}

export default App;



import React from 'react';

function App () {

  const arr = [

    {

      id:1,

      title:"Reactjs"

    },

    {

      id:2,

      title:"Nodejs"

    },

    {

      id:3,

      title:"Angularjs"

    },

    {

      id:4,

      title:"Expressjs"

    }

  ]

  return (

    <div align='center'>

      {

        arr.map(

          (value) => <li key={value.id}>{value.id}</li>

        )

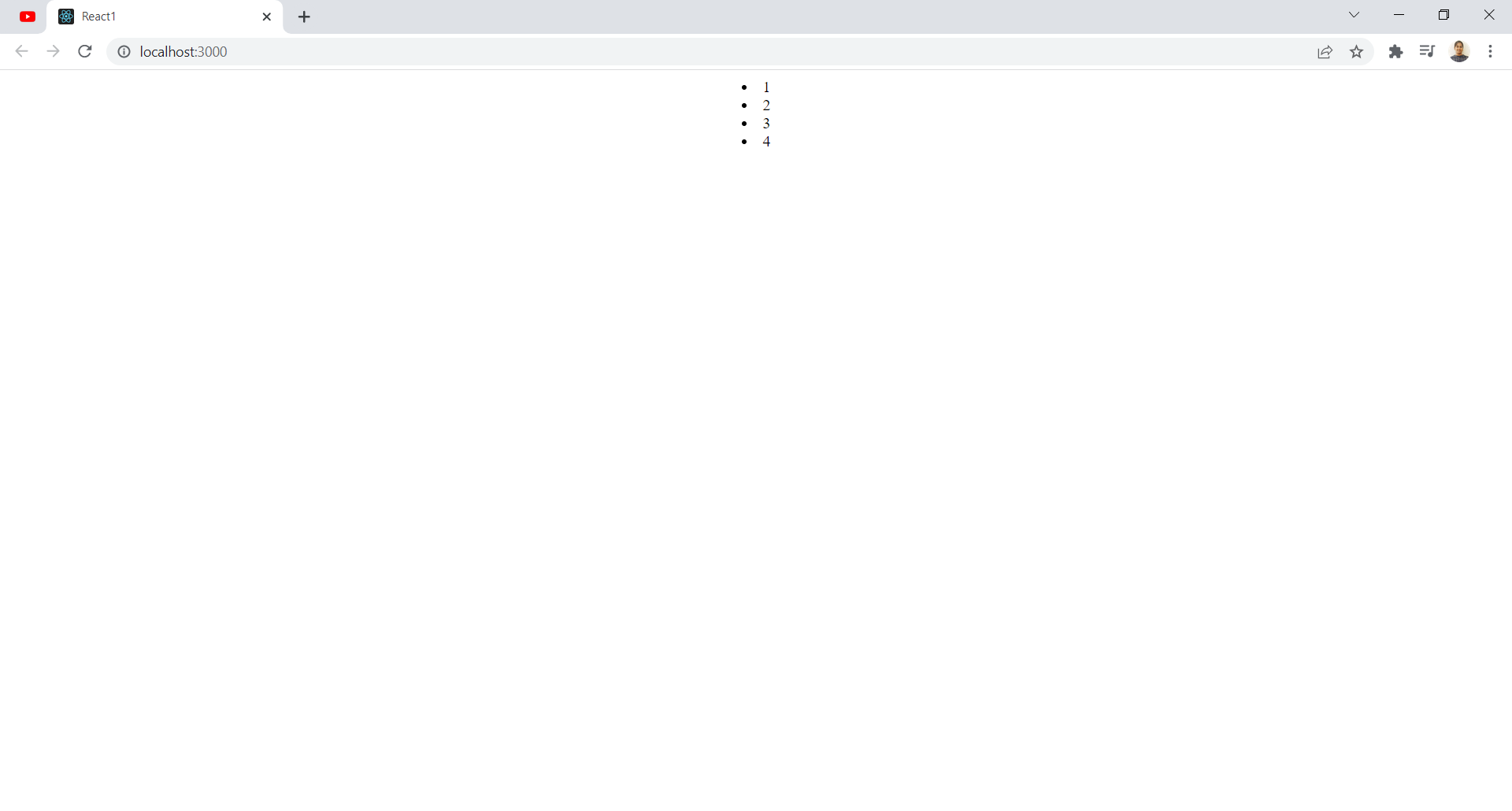
      }

    </div>

  );

}

export default App;



**Filter() function:** It is the process of looping through an array and including and excluding elements inside that array based on condition.

a)filter an array of strings.

b)filter an array of objects by value.

App.js:a)

import React from 'react';

function App(props) {

  const names = ['James' , 'John' , 'George'];

  const filtered = names.filter(name => name.includes('J'))

  return (

    <div align='center'>

      {

        filtered.map(item => <li>{item}</li>)

      }

    </div>

  );

}

export default App;



App.js:b)

import React from 'react';

function App(props) {

  const arr = [10,20,35,92,4673,872];

  const filtered = arr.filter(value => value>50)

  return (

    <div align='center'>

      {

        filtered.map(item => <li>{item}</li>)

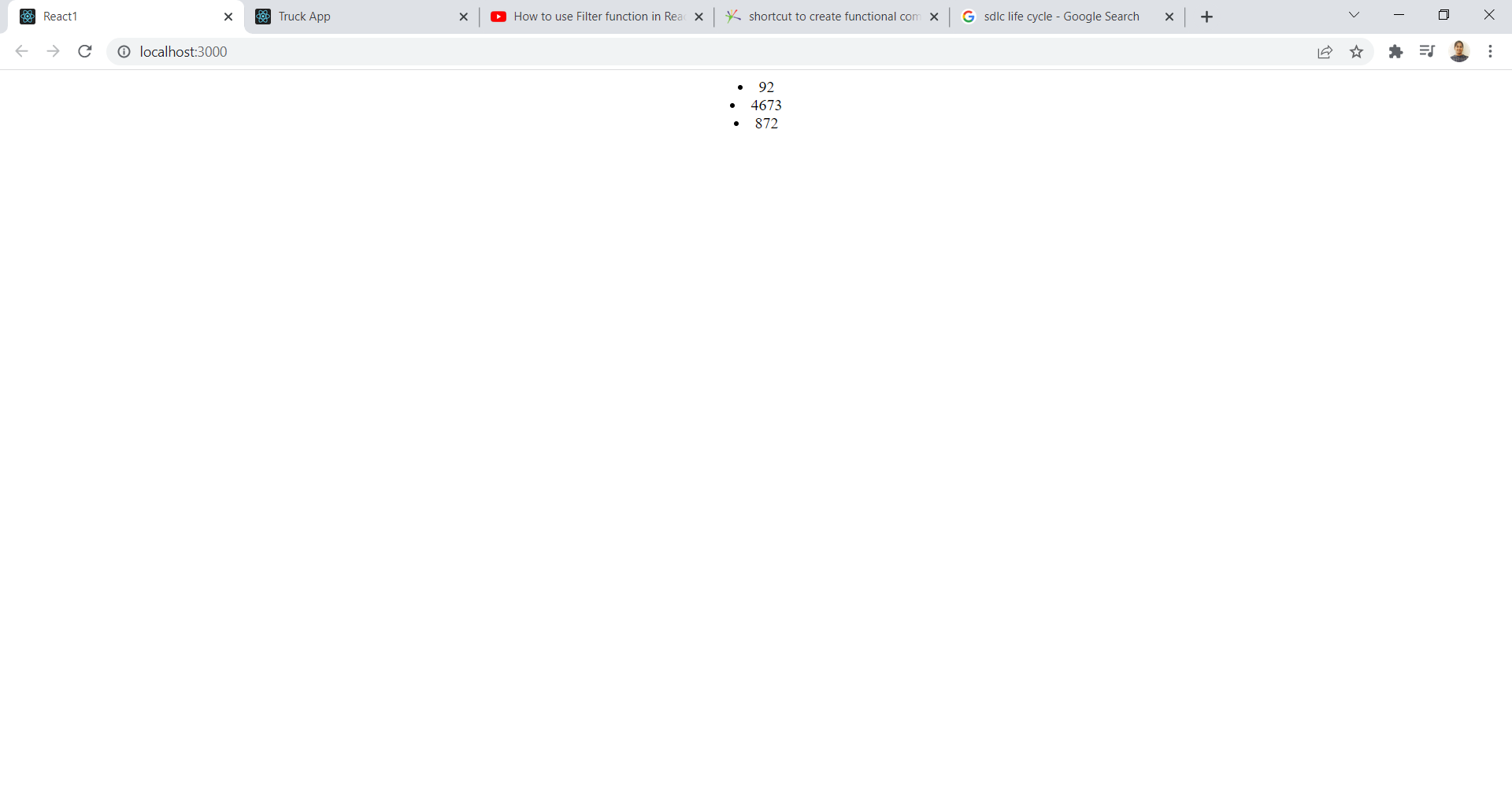
      }

    </div>

  );

}

export default App;



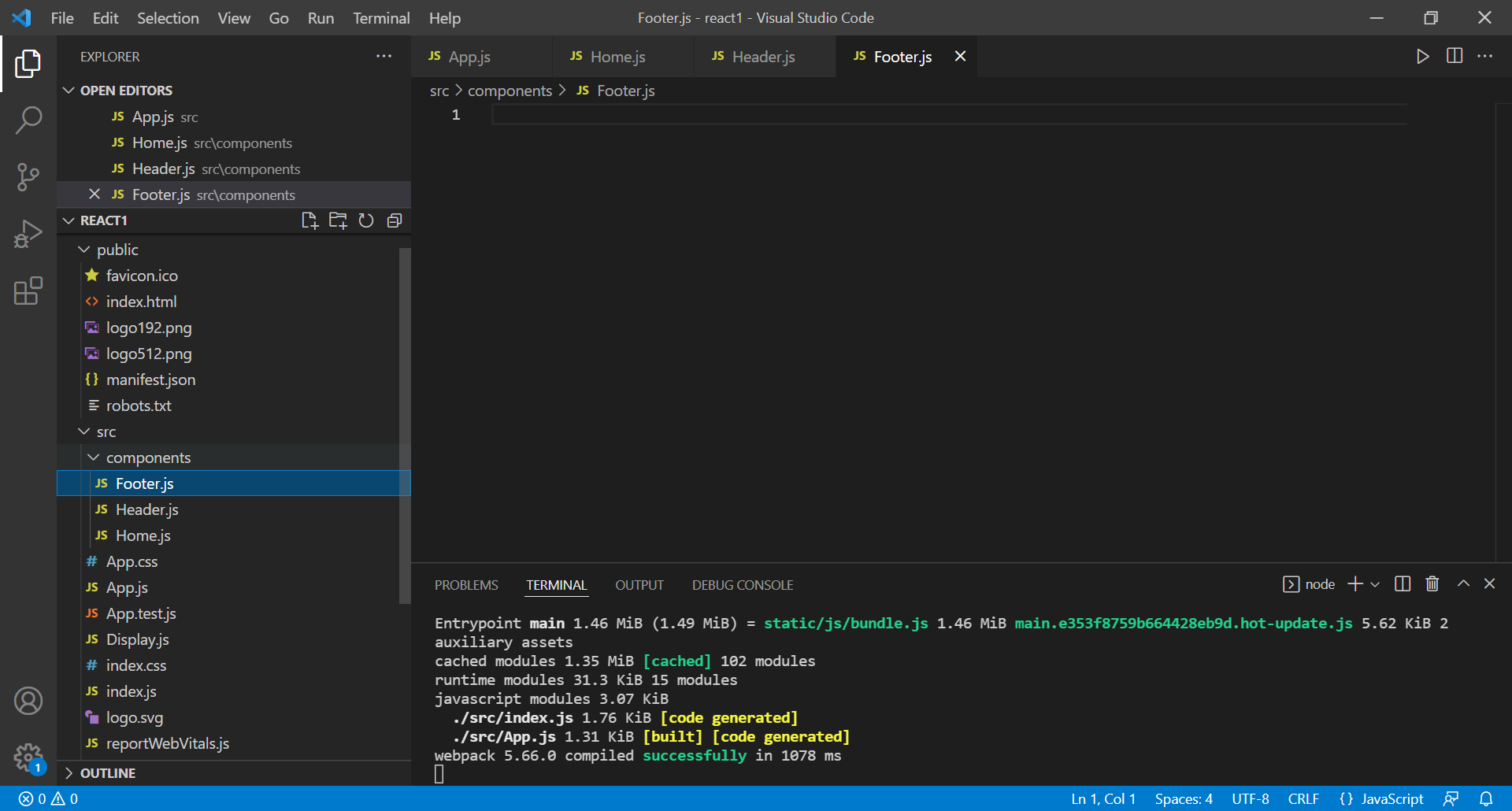
**Export vs Export default:**

1. export -export(data) =>import?
2. Default export -export(data) =>import?

**Eg:b)**

STEP-1=> create a folder in src as components.

STEP-2=>in components folder create 3 files as=>Home.js,Header.js,Footer.js



App.js:

import React from 'react';

import Home from './components/Home';

import Header from './components/Header';

import Footer from './components/Footer';

function App(props) {

  return (

    <div>

<center>

      <Home />

      <Header />

      <Footer />

</center>

    </div>

  );

}

export default App;

Home.js:

import React from 'react';

const Home = () => {

  return (

    <div>

      <h2>From HOME</h2>

    </div>

  );

}

export default Home;

Header.js:

import React from 'react';

const Header = () => {

  return (

    <div>

      <h2>From HEADER</h2>

    </div>

  );

}

export default Header;

Footer.js:

import React from 'react';

const Footer = () => {

  return (

    <div>

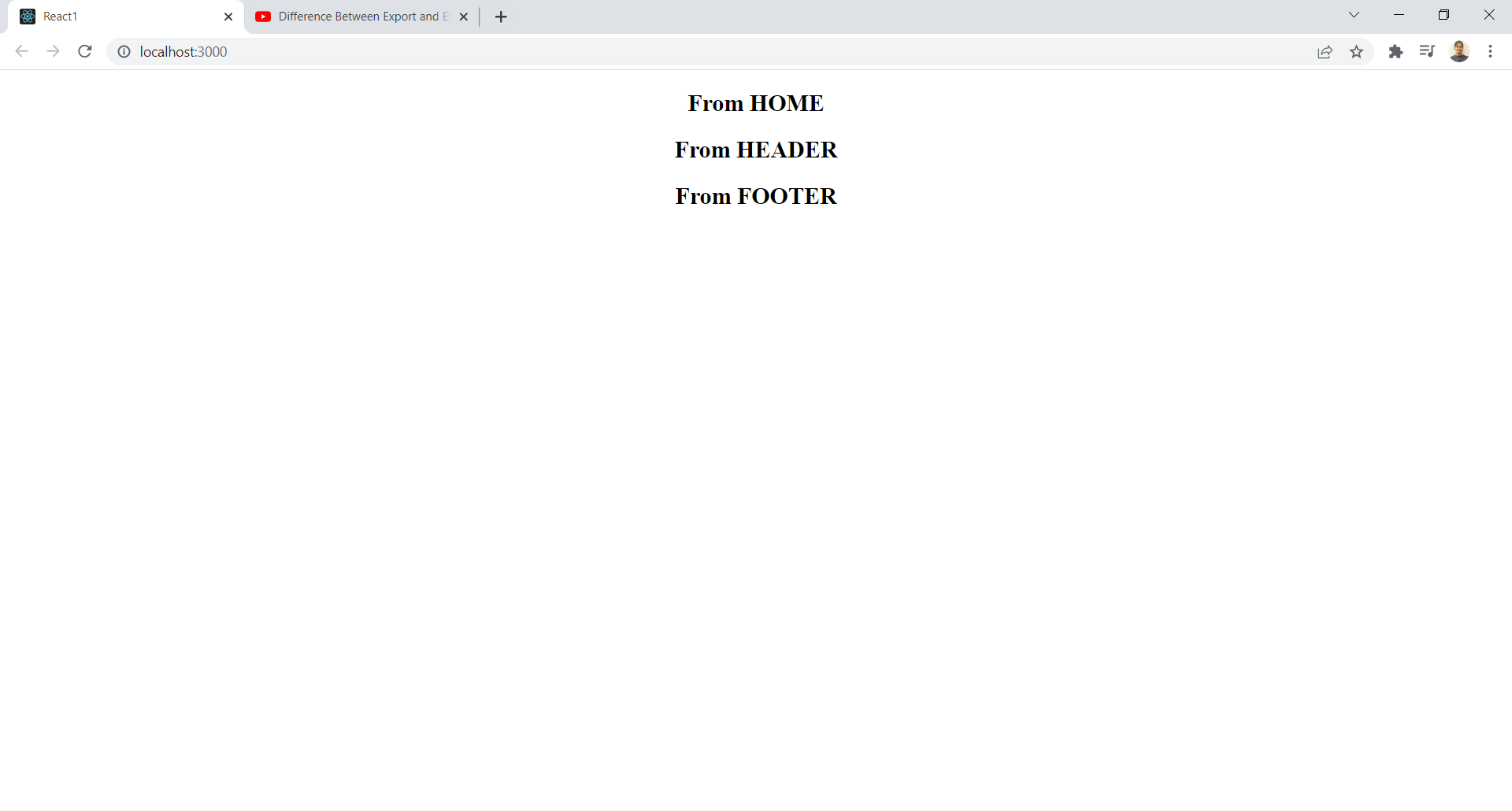
      <h2>From FOOTER</h2>

    </div>

  );

}

export default Footer;



**Eg:a)** (same as above eg,BUT we use curly braces for component name in App.js ,export word before const in Home,Header and Footer and don’t use export default App; in Home,Header and Footer.)

App.js:

import React from 'react';

import {Home} from './components/Home';

import {Header} from './components/Header';

import {Footer} from './components/Footer';

function App(props) {

  return (

    <div>

      <Home />

      <Header />

      <Footer />

    </div>

  );

}

export default App;

Home.js:

import React from 'react';

export const Home = () => {

  return (

    <div>

      <h2>From HOME</h2>

    </div>

  );

}

Header.js:

import React from 'react';

export const Header = () => {

  return (

    <div>

      <h2>From HEADER</h2>

    </div>

  );

}

Footer.js:

import React from 'react';

export const Footer = () => {

  return (

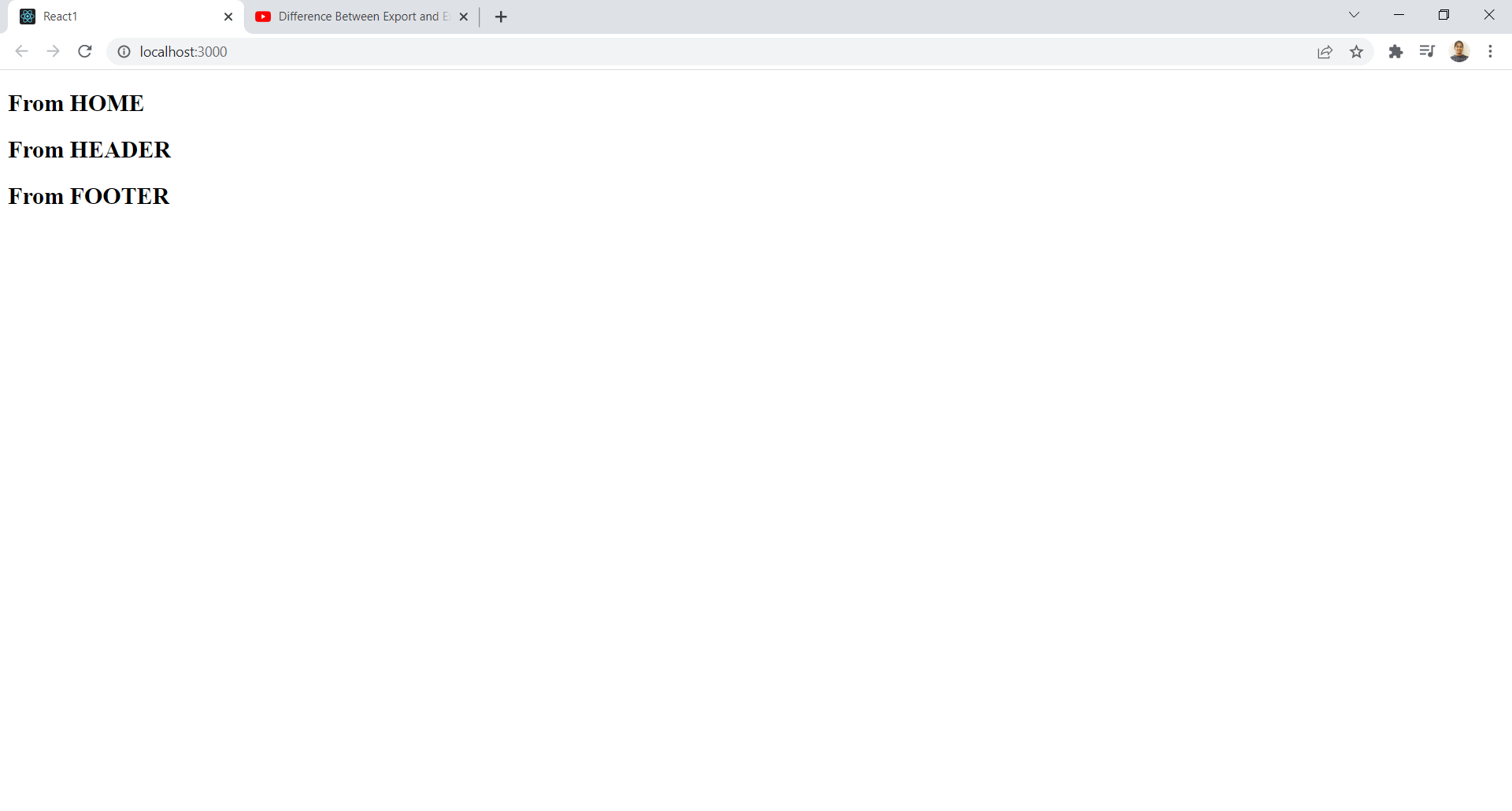
    <div>

      <h2>From FOOTER</h2>

    </div>

  );

}



**LOGIN FORM SUBMIT:**

Key concepts:

a)onClick,onChange,onSubmit

b)useState()-ReactHook

c)object destructuring

**SIGNUP FORM:**

Key concepts:

a)onClick,onChange,onSubmit

b)useState()-ReactHook

c)object destructuring

Eg:

import React , {useState} from 'react';

const App = () => {

  const [data,setData] = useState({

    username : '',

    email : '',

    password :'',

    confirmPassword : ''

  })

  const {username,email,password,confirmPassword} = data;

  const changeHandler = e =>{

    setData({...data,[e.target.name]:e.target.value})

  }

  const submitHandler = e => {

    e.preventDefault();

    if(password === confirmPassword)

    {

      console.log(data);

    }

    else{

      console.log("Passwords are not matched");

    }

  }

  return (

    <div>

    <center>

    <form onSubmit={submitHandler}>

      <input type="text" name="username" placeholder="enter name" value={username} onChange={changeHandler} /><br/>

      <input type="email" name="email" placeholder="enter mail id" value={email} onChange={changeHandler} /><br/>

      <input type="password" name="password" placeholder="enter password" value={password} onChange={changeHandler} /><br/>

      <input type="confirmPassword" name="confirmPassword" placeholder="re-enter the password" value={confirmPassword} onChange={changeHandler} /><br/>

      <input type="submit" name="SUBMIT"/><br/>

    </form>

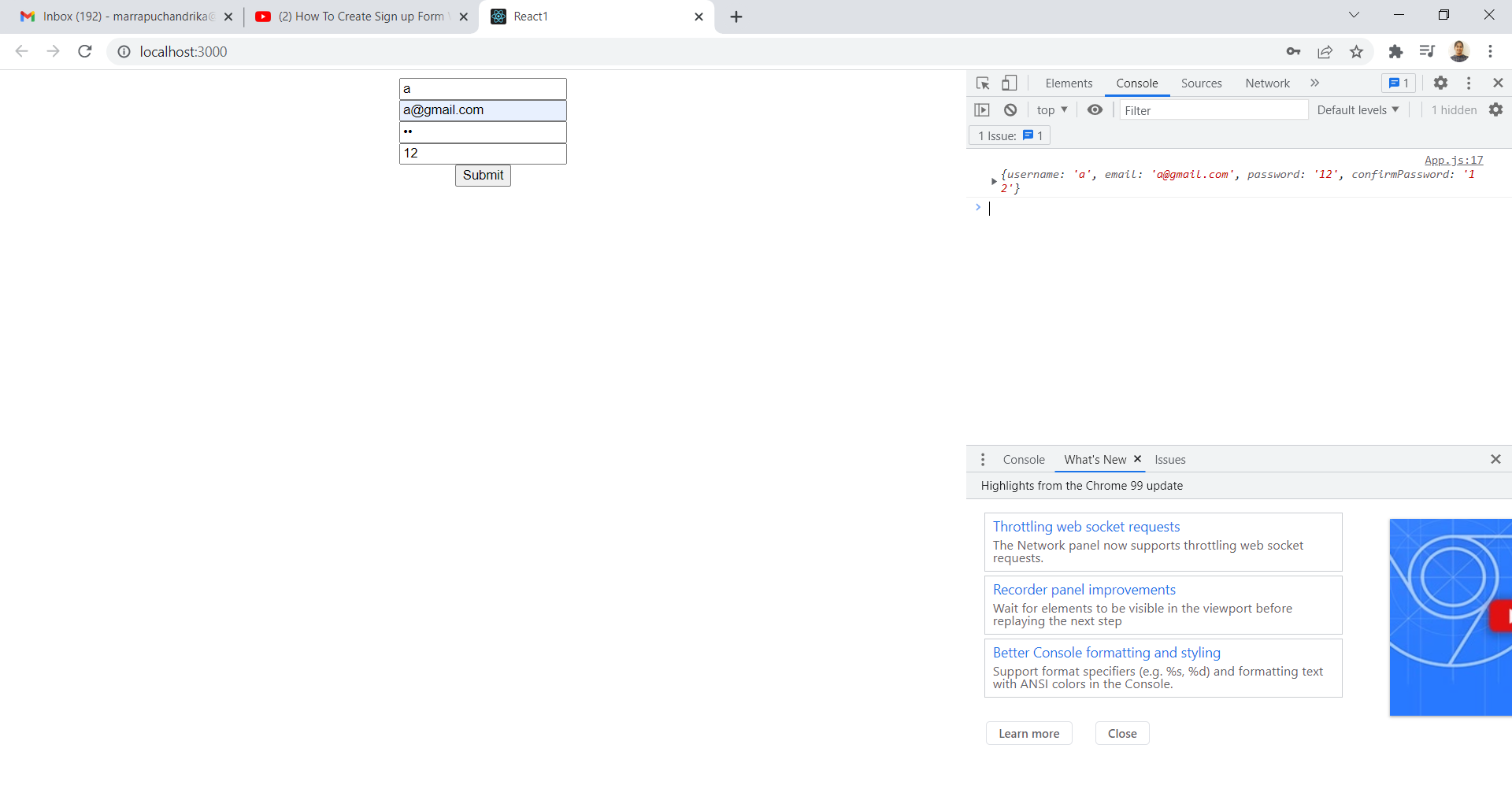
    </center>

    </div>

  );

}

export default App;



**FORM VALIDATION:** To check user given input data is valid or not and sends to database.

Eg: SIGNUP form with VALIDATION

import React , {useState} from 'react';

const App = () => {

  const [data,setData] = useState({

    username : '',

    email : '',

    password :'',

    confirmPassword : ''

  })

  const {username,email,password,confirmPassword} = data;

  const changeHandler = e =>{

    setData({...data,[e.target.name]:e.target.value})

  }

  const submitHandler = e => {

    e.preventDefault();

    if(username.length <= 5)

    {

      alert("username must be morethan 5 characters");

    }

    else if(password !== confirmPassword){

      alert("passwords are not matching");

    }

    else{

      console.log(data)

    }

  }

  return (

    <div>

    <center>

    <form onSubmit={submitHandler}>

      <input type="text" name="username" placeholder="enter name" value={username} onChange={changeHandler} /><br/>

      <input type="email" name="email" placeholder="enter mail id" value={email} onChange={changeHandler} /><br/>

      <input type="password" name="password" placeholder="enter password" value={password} onChange={changeHandler} /><br/>

      <input type="confirmPassword" name="confirmPassword" placeholder="re-enter the password" value={confirmPassword} onChange={changeHandler} /><br/>

      <input type="submit" name="SUBMIT"/><br/>

    </form>

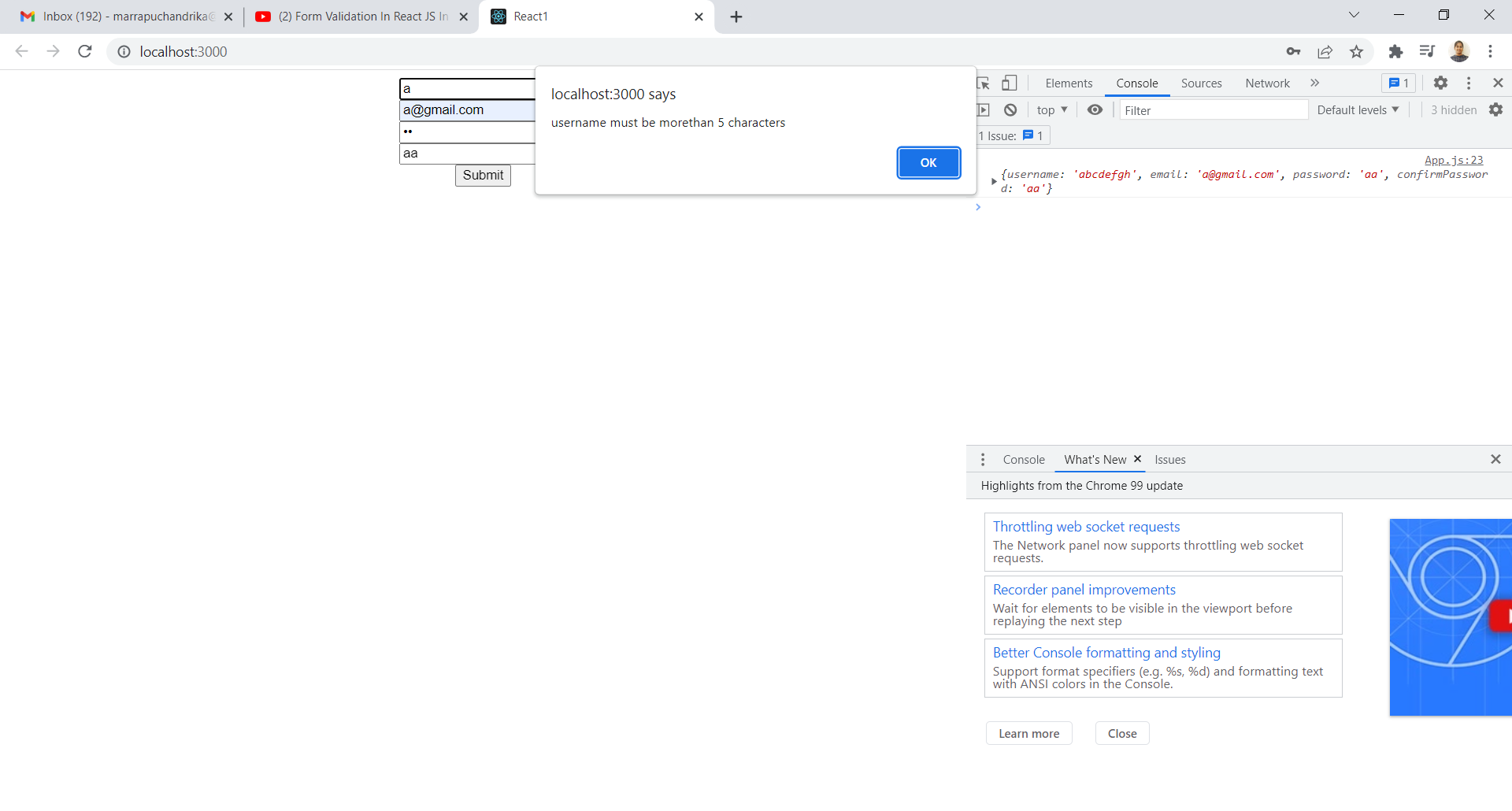
    </center>

    </div>

  );

}

export default App;



**How to get API data using Fetch API and Axios:**

**a)Fetch API:**

App.js

import React,{useEffect} from 'react';

const App = () => {

  useEffect(() => {

  fetch('http://jsonplaceholder.typicode.com/todos').then(

    response => response.json()

  ).then(json => console.log(json))

},[])

  return (

    <div>

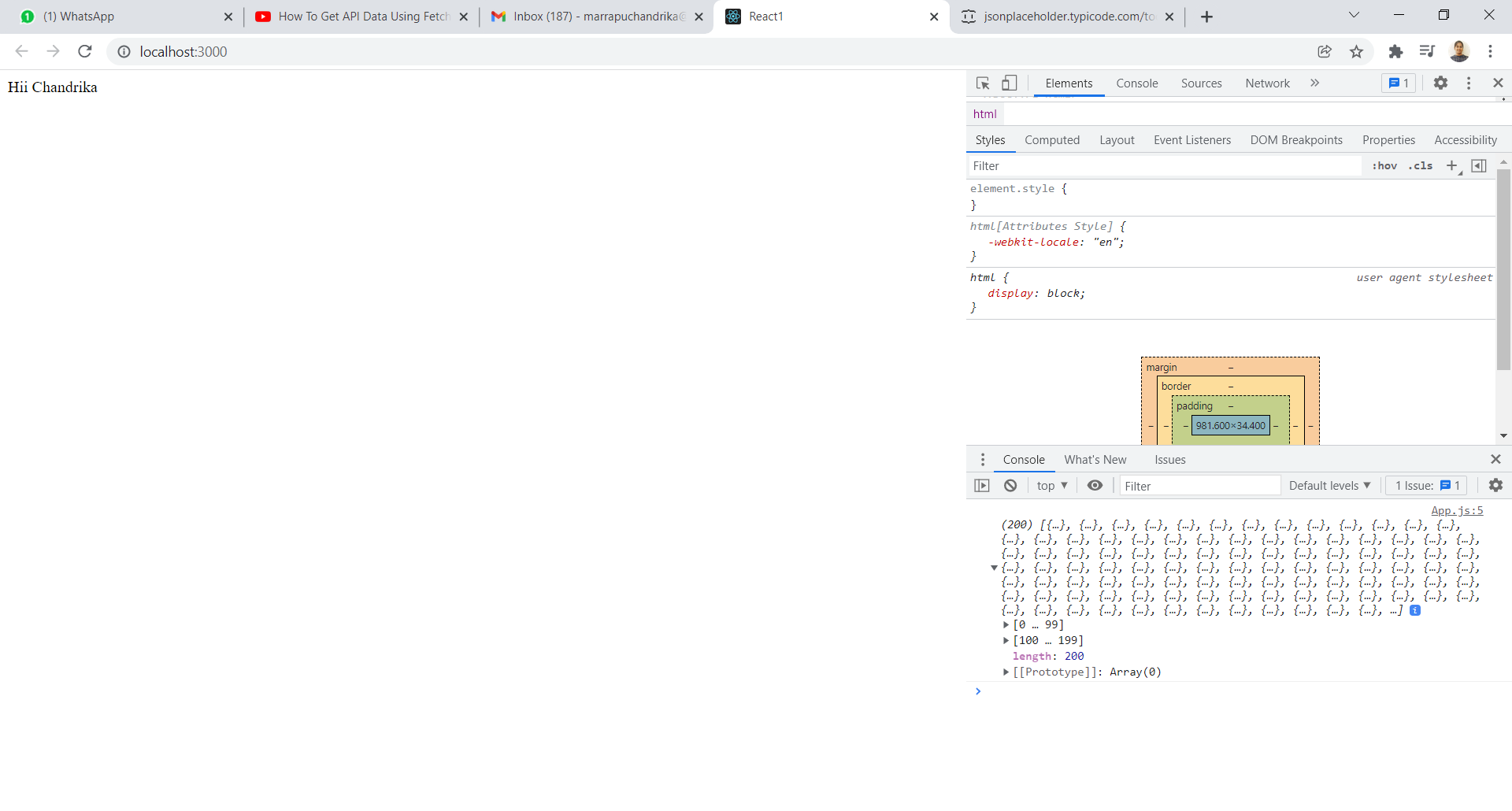
      Hii Chandrika

    </div>

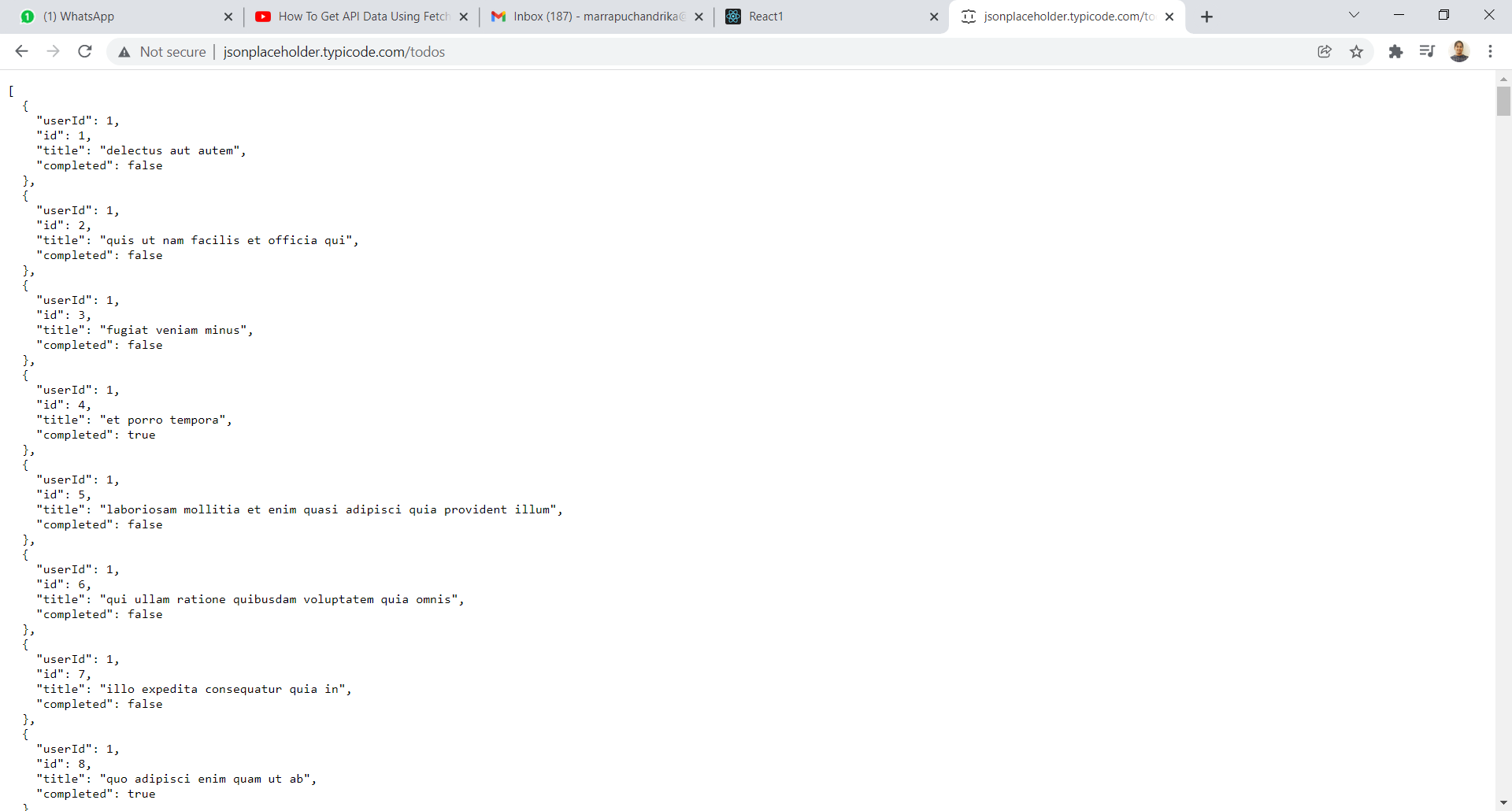
  );

}

export default App;



Input data: 'http://jsonplaceholder.typicode.com/todos



The above example is for fetching data – How to use in React project?

ANS: Store in variable by using React Hook.

import React,{useEffect,useState} from 'react';

const App = () => {

  const [data,setData] = useState([]);

    useEffect(() => {

  fetch('http://jsonplaceholder.typicode.com/todos').then(

    response => response.json()

  ).then(json => setData(json))

},[])

  return (

    <div>

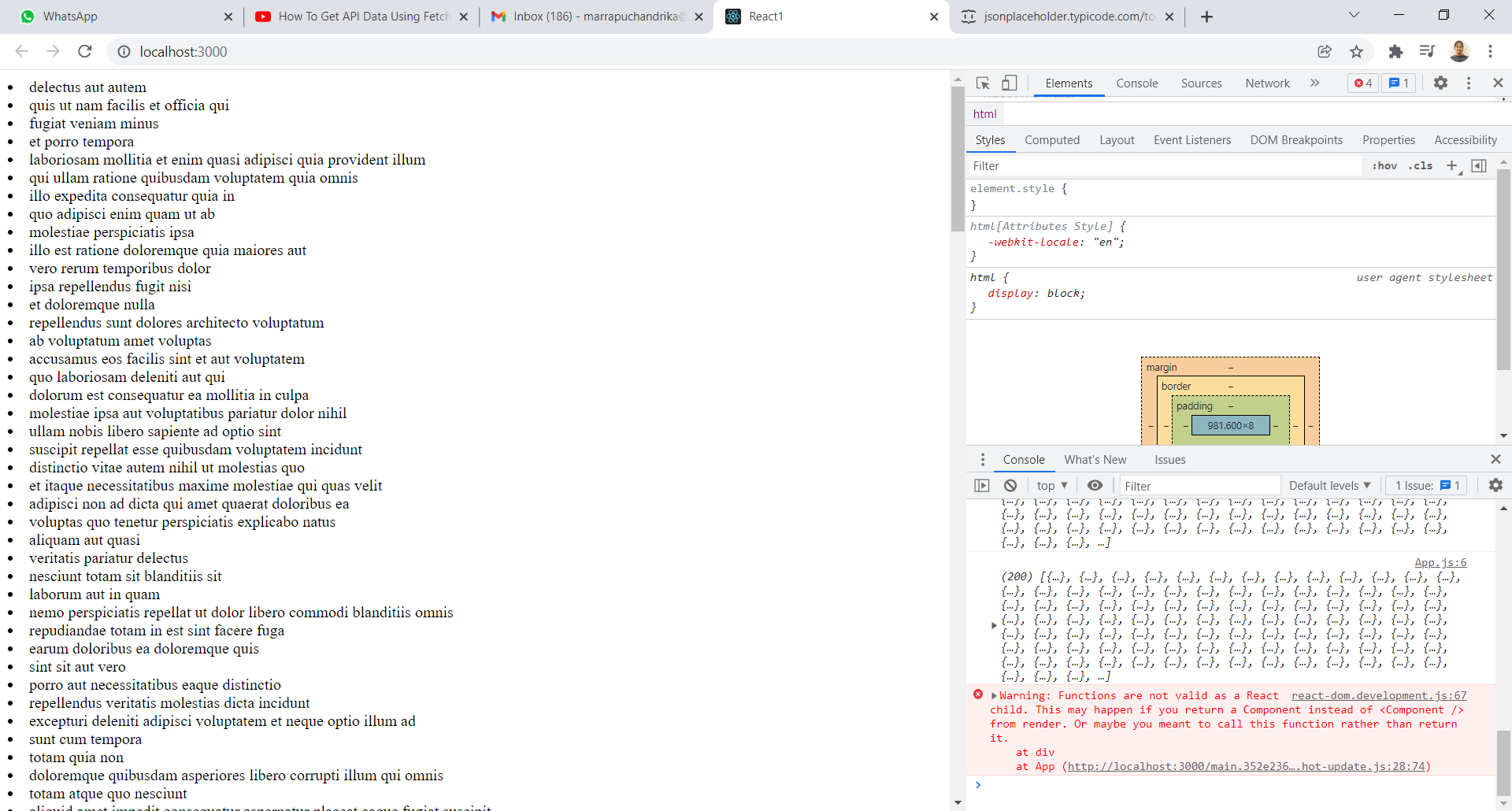
      {data.map(item => <li key={item.id}>{item.title}</li>)}

    </div>

  );

}

export default App;



**b)Axios:** It is library ,we can make any type of http request ,like getrequest,postrequest and delete request.

npm install axios(in react1 folder)

import axios from 'axios';

import React,{useEffect,useState} from 'react';

const App = () => {

  const [data,setData] = useState([]);

    useEffect(() => {

    axios.get("http://jsonplaceholder.typicode.com/todos").then(

    response => setData(response.data)

  )

},[])

  return (

    <div>

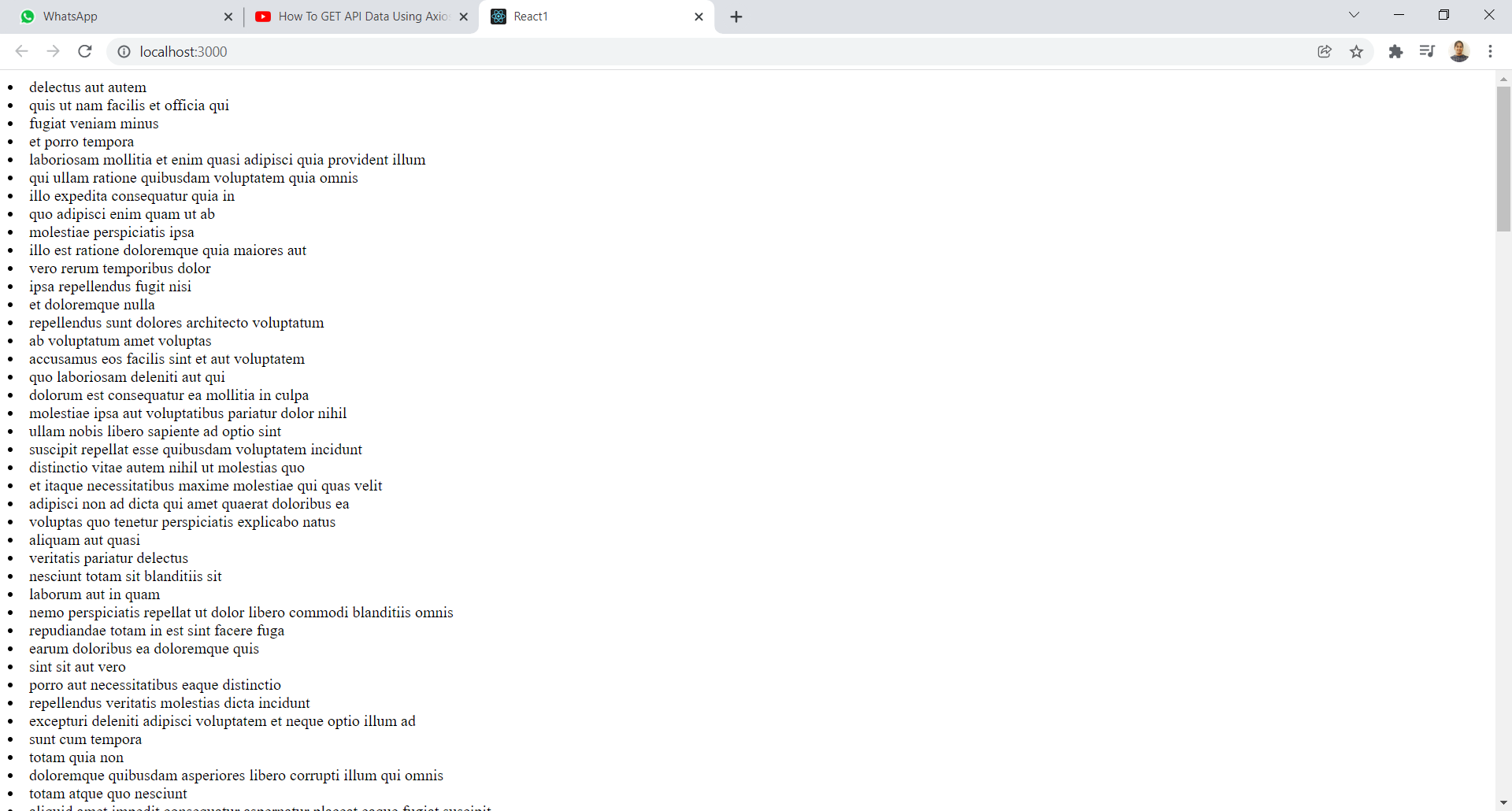
      {data.map(item => <li key={item.id}>{item.title}</li>)}

    </div>

  );

}

export default App;

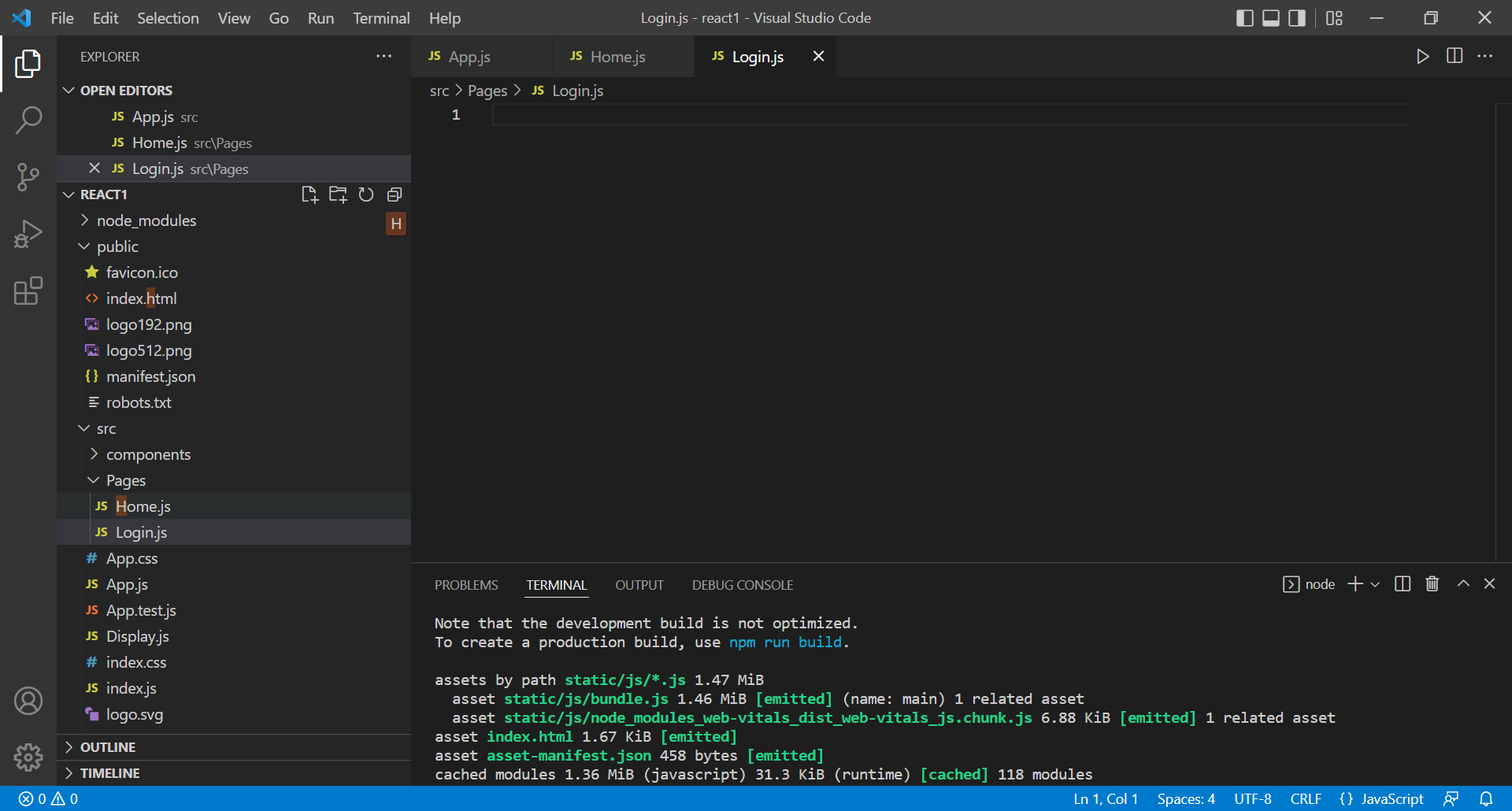


INCEDO-Reactjs

HOME page and LOGIN page:[Validation check]

1)Add a folder under src.

Src>new folder>pages(create)-HOME file and login file(create)



Install package: npm install react-router-dom

