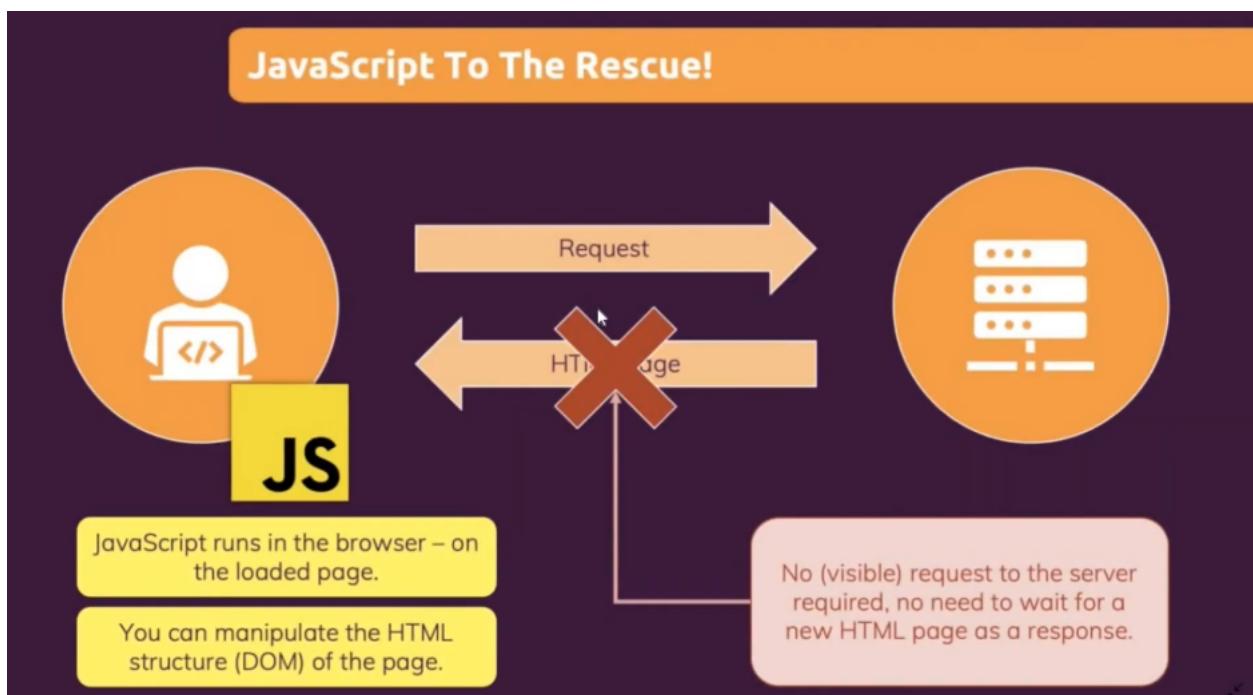


What is ReactJs?

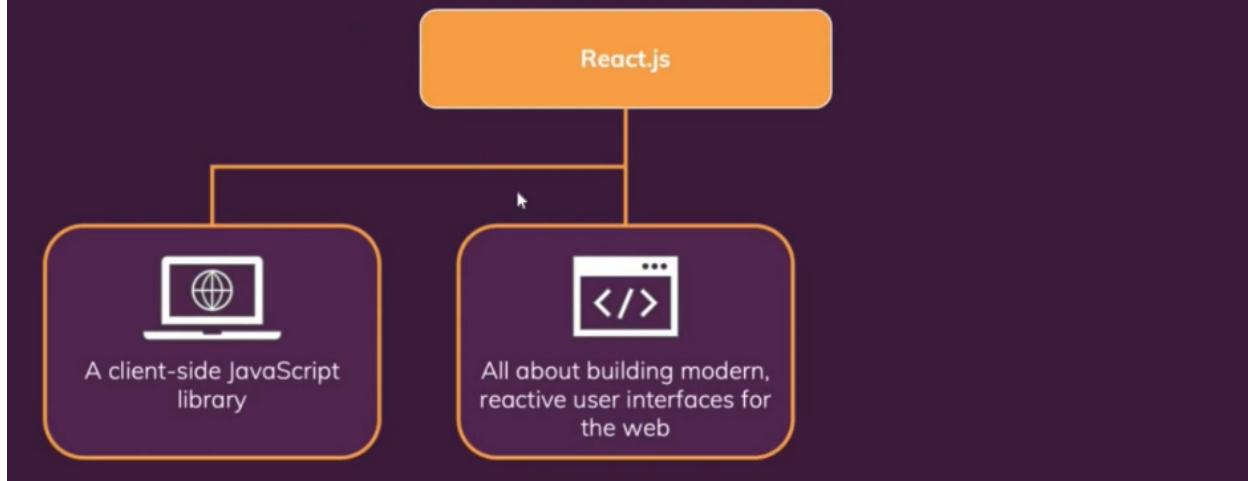
React is a JavaScript library for building user interfaces

→ JavaScript (no need to run server on multiple times for every request)

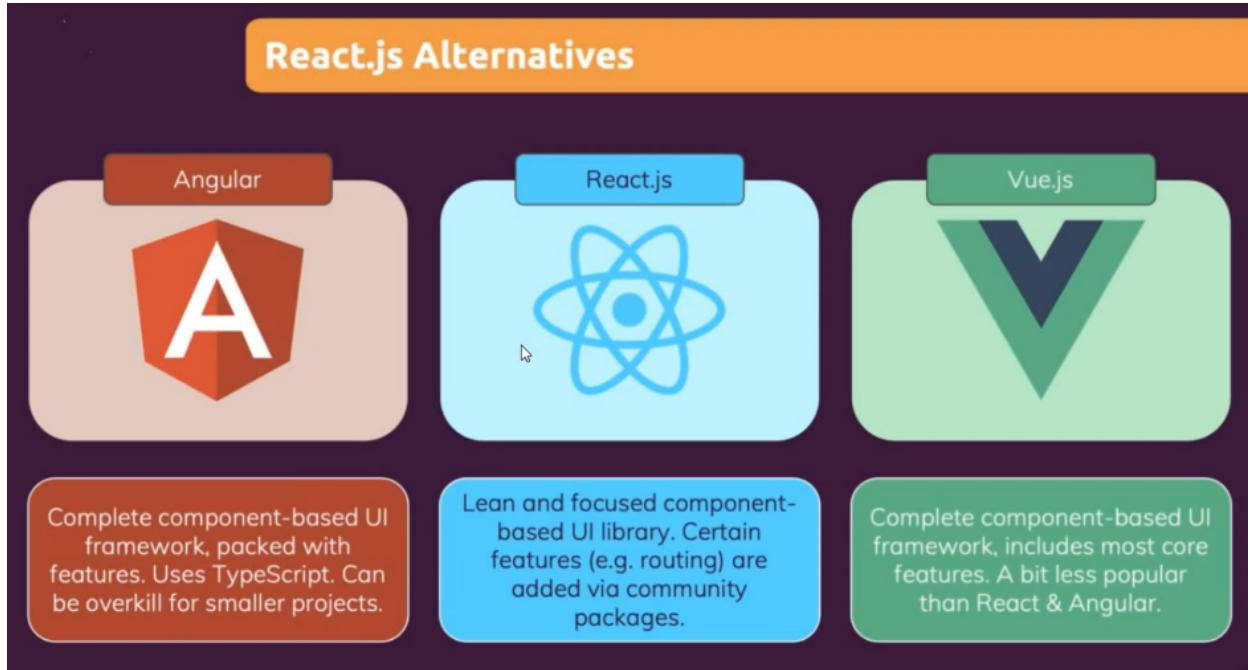


- In react.js we need to install packages from “npm”
- Vue.js is less popular
- In angular.js the packages are already inbuilt & no need to install.

What is React.js?



React.js Alternatives



How to Use Arrow Function in JavaScript

newVersion →ES6

```
function oldversion(){
    console.log("oldVersion");
}
oldversion(); //call function

//new version arrow function
const newversion = () =>{
    console.log("newVersion");
}
newversion(); //call function
```

O/P:

```
PS C:\Users\gudiw> node sample.js
oldVersion
newVersion
```

```
function oldversion(number1){
    return number1 + 60;
}
console.log(oldversion(50));

//new version arrow function
const newversion = (num1, num2) =>{
    return num1 + num2;
}
console.log(newversion(10, 20));
```

```
//advanced reduce no of lines
const newversion1 = (num1, num2) => num1 + num2;

console.log(newversion1(100, 20));

//otherway single parameter no need to use brackets
const newversion2 = num1 => num1 + 78;

console.log(newversion2(10));
```

O/P:

PS C:\Users\gudiw> **node sample.js**

110

30

120

88

ES6 JavaScript Classes

ReferenceError: Must **call super constructor** in derived class before accessing 'this' or returning from derived constructor

```
class human{  
    constructor(){  
        this.age = 25;  
    }  
    printAge(){      // Method  
        console.log(this.age);  
    }  
}  
  
class person extends human{  
    constructor(){  
        super();  
        this.name = "abc";  
    }  
    printName(){  
        console.log(this.name);  
    }  
}  
  
const p1 = new person(); //obj for child class  
p1.printName();  
p1.printAge();
```

O/P:

```
PS C:\Users\gudiw> node sample.js
```

```
abc
```

```
25
```

```
class human{  
    constructor(){  
        this.age = 25;  
    }  
    printAge(){      // Method  
        console.log(this.age);  
    }  
}  
  
class person extends human{  
    constructor(){  
        super();  
        this.name = "abc";  
        this.age = 40;      // Updating the age  
    }  
    printName(){  
        console.log(this.name);  
    }  
}  
  
const p1 = new person(); //obj for child class  
p1.printName();  
p1.printAge();
```

O/P:

```
PS C:\Users\gudiw> node sample.js
abc
40
```

ES7 JavaScript Classes

```
class human{

    age = 25;
    printAge = () => console.log(this.age);
}

class person extends human{

    name = "abc";
    age = 40;          // Updating the age
    printName = () => console.log(this.name);
}

const p1 = new person();    //obj for child class
p1.printName();
p1.printAge();
```

O/P:

```
PS C:\Users\gudiw> node sample.js
abc
40
```

Spread Operator in JavaScript

Spread Operator (...):

- The spread operator is used to expand or spread the elements of an array or object. It allows you to clone or combine arrays, or add new elements to an existing array.
- When used with an array, it can be used to create a new array by expanding the elements of an existing array.
- It can also be used to clone an array or pass elements of an array as arguments to a function.

```
set1 = [1,2,3,4,5];
set2 = [...set1,6,7,8,9,10];

console.log(set2);
```

O/P:

```
PS C:\Users\gudiw> node sample.js
[
  1, 2, 3, 4, 5,
  6, 7, 8, 9, 10
]
```

```
set1 = {
  name: "abc",
  email: "rag123@gmail.com"
}

set2 = {
  ...set1,
```

```
    age: 25
}

console.log(set2);
```

O/P:

```
PS C:\Users\gudiw> node sample.js
{ name: 'abc', email: 'rag123@gmail.com', age: 25 }
```

Destructuring in JavaScript

→ Array destructuring

```
values = [10, 20];

[a, b] = values;

console.log(a, b);

values = [10, 20, 30];

[a, b, c] = values;

console.log(a, b, c);

values = [10, 20, 30];

[a, , c] = values;

console.log(a, c);
```

O/P:

PS C:\Users\gudiw> **node sample.js**

10 20

10 20 30

10 30

→**Without destructuring**

```
values = {  
    email: "rag123@gmail.com",  
    age: 25  
}  
  
console.log(values.email);
```

O/P:

PS C:\Users\gudiw> **node sample.js**

rag123@gmail.com

→**Object Destructuring**

```
values = {  
    email: "rag123@gmail.com",  
    age: 25  
}  
  
//const {email, age} = values;  
const {age, email} = values;  
  
console.log(email);
```

O/P:

PS C:\Users\gudiw> **node sample.js**
rag123@gmail.com

How to Use Rest Operator in JavaScript

Rest Operator (...):

- The rest operator is used to represent multiple elements as an array. It allows you to capture a variable number of arguments passed to a function or the remaining elements in an array.
- When used in function parameters, the rest operator allows a function to accept any number of arguments as an array.
- It can also be used to capture the remaining elements of an array during destructuring.

```
const array = (...arr) => console.log(arr);

array(10,20,30,40,50,60);

const array1 = (a, ...arr) => console.log(arr);

array1(10,20,30,40,50,60);

const array2 = (a, b, ...arr) => console.log(a, b, arr);

array2(10,20,30,40,50,60);
```

O/P:

```
PS C:\Users\gudiw> node sample.js
```

```
[ 10, 20, 30, 40, 50, 60 ]
```

```
[ 20, 30, 40, 50, 60 ]
```

```
10 20 [ 30, 40, 50, 60 ]
```

In summary, the rest operator is used to capture multiple elements as an array, while the spread operator is used to expand or spread the elements of an array or object.

Array in JavaScript

→ In javascript, we can store **heterogeneous data** in an array.

array.push()

Use pop() to remove from end

Use shift() to remove from start

Use splice() to remove arbitrary

The **splice()** method is used to modify an array by adding, removing, or replacing elements within the array. It provides a flexible way to manipulate arrays.

array.splice(startIndex, deleteCount, item1, item2, ...);

- **startIndex**: The index at which the modification should begin.
- **deleteCount** (optional): The number of elements to remove starting from the **startIndex**. If **deleteCount** is 0, no elements are removed.
- **item1, item2, ...** (optional): The elements to add to the array starting from the **startIndex**. If no items are specified, only removal of elements will occur.

The **splice()** method returns an array containing the removed elements, or an empty array if no elements were removed.

Example:

```
const numbers = [1, 2, 3, 4, 5];

// Remove elements from the array
numbers.splice(2, 2); // Remove 2 elements starting from index 2
console.log(numbers); // Output: [1, 2, 5]

// Add elements to the array
numbers.splice(2, 0, 6, 7); // Insert 6 and 7 at index 2
console.log(numbers); // Output: [1, 2, 6, 7, 5]

// Replace elements in the array
numbers.splice(1, 1, 8, 9); // Replace 1 element at index 1 with 8 and 9
console.log(numbers); // Output: [1, 8, 9, 6, 7, 5]

// Retrieve removed elements
const removedElements = numbers.splice(1, 3);
console.log(removedElements); // Output: [8, 9, 6]
console.log(numbers); // Output: [1, 7, 5]
```

The `splice()` method provides powerful array manipulation capabilities, allowing you to modify an array in place.

```
let arr = [10,20,30,40]
console.log(arr);

arr.push(50);
console.log(arr);

arr.pop();
console.log(arr);

arr.shift();
console.log(arr);

let index = 2;
arr.splice(index, 1); //1 indicate no of values to be deleted
console.log(arr);

let index1 = arr.indexOf(20);
arr.splice(index1, 1);
console.log(arr);
```

O/P:

```
PS C:\Users\gudiw> node sample.js
[ 10, 20, 30, 40 ]
[ 10, 20, 30, 40, 50 ]
[ 10, 20, 30, 40 ]
[ 20, 30, 40 ]
[ 20, 30 ]
[ 30 ]
```

How to Create ReactJs App

Module Content

React Core Syntax & JSX

Working with Components

Working with Data

Reactjs requirement → nodejs, VS Code

`npm install create-react-app`

`npx create-react-app project_name`

`npm start`

→I have already installed “`npm install create-react-app`”

C:\Users\gudiw\Documents\tuts>**npx create-react-app sample**

Creating a new React app in C:\Users\gudiw\Documents\tuts\sample.

Installing packages. This might take a couple of minutes.

Installing react, react-dom, and react-scripts with cra-template...

added 1413 packages in 6m

226 packages are looking for funding

run `npm fund` for details

Initialized a git repository.

Installing template dependencies using npm...

added 74 packages, and changed 1 package in 4m

235 packages are looking for funding

run `npm fund` for details

Removing template package using npm...

removed 1 package in 9s

235 packages are looking for funding

run `npm fund` for details

Created git commit.

Success! Created sample at C:\Users\gudiw\Documents\tuts\sample

Inside that directory, you can run several commands:

npm start

Starts the development server.

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 sample  
npm start
```

Happy hacking!

C:\Users\gudiw\Documents\tuts>**cd sample**

C:\Users\gudiw\Documents\tuts\sample>**code .** //This will open VS Code

Type >> **npm start** in VS Code we get below results

Compiled successfully!

You can now view sample in the browser.

Local: **http://localhost:3000**

On Your Network: **http://192.168.0.100:3000**

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

webpack compiled successfully

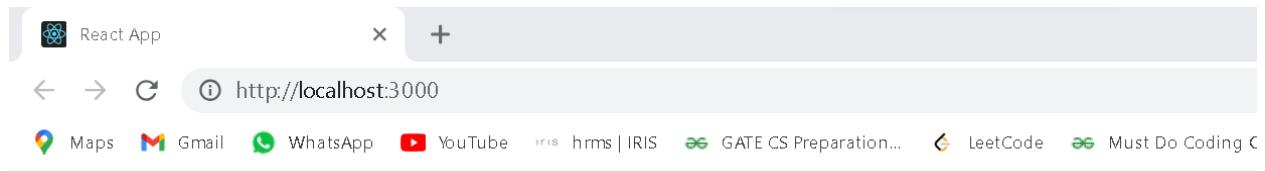
App.js

```
import React from "react";

const App = () => {
  return (
    <div>
      <h1>Hello Word !!</h1>
    </div>
  );
}

export default App;
```

O/P:



Hello Word !!

Function and Class Component in ReactJs

<https://github.com/rodrigovallades/vscode-es7-javascript-react-snippets>

ES7 React/Redux/GraphQL/React-Native snippets →install

→JavaScript code is to be used in html. We use curly brackets.

→In class we use “render”

Class Level Component

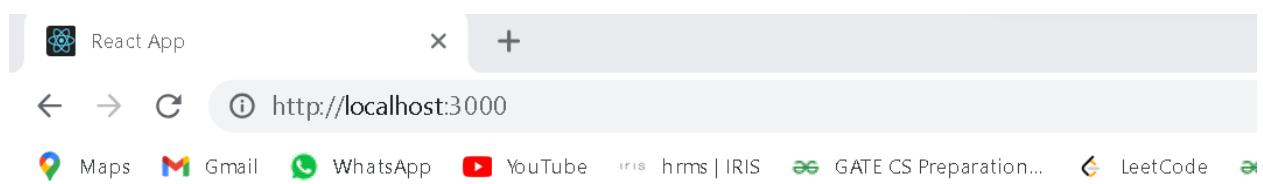
“rcc” using the extension

```
import React, { Component } from 'react'

export default class App extends Component {
  state = {
    name: "Telugu Skillhub",
  }
  render() {
    return (
      <div>
        <h1>{this.state.name}</h1>
      </div>
    )
  }
}
```

```
<div>
  <h2>Welcome to { this.state.name}</h2>
</div>
)
}
}
```

O/P:



Welcome to Telugu Skillhub

Function Level Component

→ React hook “`useState`” is used only in Function level component

“rafc” using the extension

```
import React, {useState} from 'react'

const App = () => {
  const [name, setName] = useState("Telugu Skillhub !!!");

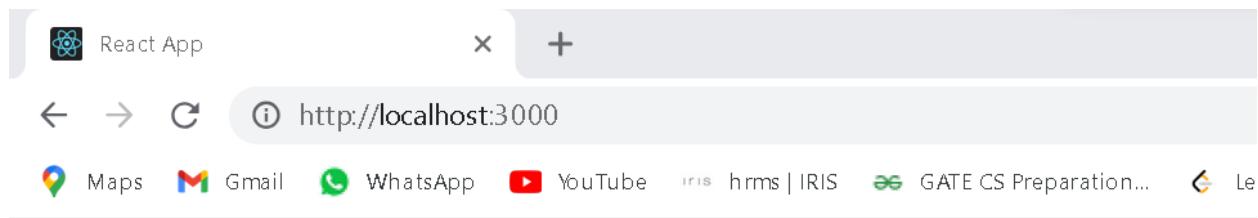
  return (
    <div>
      <h2>Welcome to {name} </h2>
    </div>
  );
}

export default App;
```

```
        </div>
    )
}

export default App
```

O/P:



Welcome to Telugu Skillhub !!!

→ After 16.8 mostly they are using Function level component

State and Props in Class Level Component

Props → send values from one component to other component

State → In one component we need to declare multiple variable values

→ From App.js we are send name to to the Display.js (through “props”)

App.js

```
import React, { Component } from 'react'
import Display from './Display'

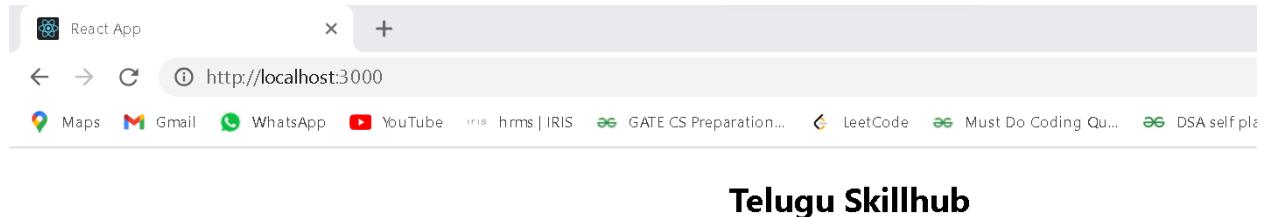
export default class App extends Component {
```

```
state={  
  name: "Telugu Skillhub"  
}  
  
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>{this.props.name}</h2>  
      </div>  
    )  
  }  
}
```

O/P:



How to Use CSS Styling in ReactJs

Div Component

. → Class
→ Id

External

App.js

```
import React from 'react'
import './App.css';

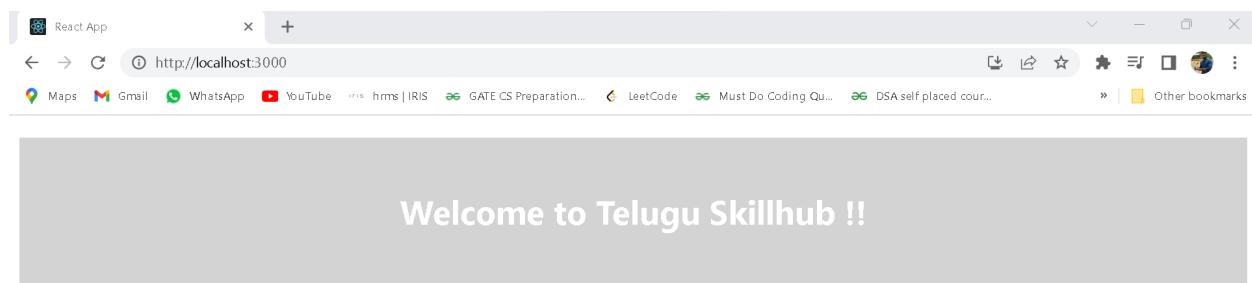
const App = () => {
  return (
    <div className='container'>
      <h1>Welcome to Telugu Skillhub !!</h1>
    </div>
  )
}

export default App
```

App.css

```
h1{  
    color: white;  
    background-color: lightgray;  
    padding: 50px;  
    text-align: center;  
}  
  
.container{  
    margin: 10px;  
}
```

O/P:



Internal → Inline

- In HTML you have to use CSS we need two Brackets {}{}}
- In HTML you have to use JavaScript we need Single Bracket { }}

App.js

```
import React from 'react'  
// import './App.css';  
  
const App = () => {  
    return (  
        <h1>Welcome to Telugu Skillhub !!</h1>  
    );  
};  
  
export default App;
```

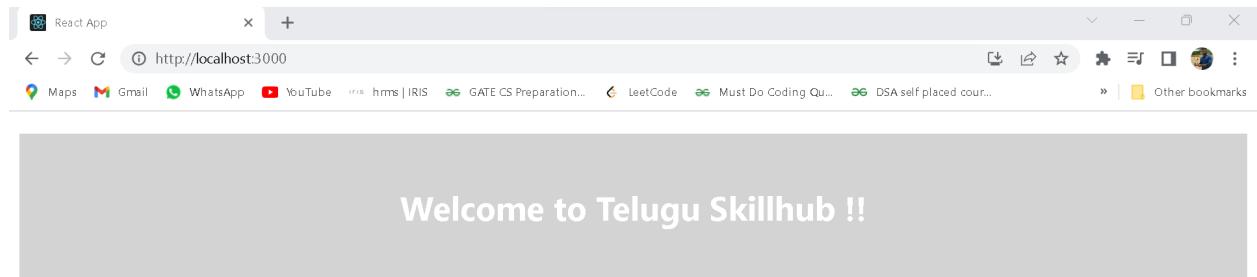
```

        <div style= {{margin:"10px"}}>
          <h1 style={{color:"white",backgroundColor:
"lightgray",padding: "50px",textAlign: "center"}}>Welcome to
Telugu Skillhub !!</h1>
        </div>
      )
}

export default App

```

O/P:



→looks good for user modifying the code

```

import React from 'react'

const App = () => {
  const styling = {color:"white",
    backgroundColor: "lightgray",
    padding: "50px",
    textAlign: "center"
  }

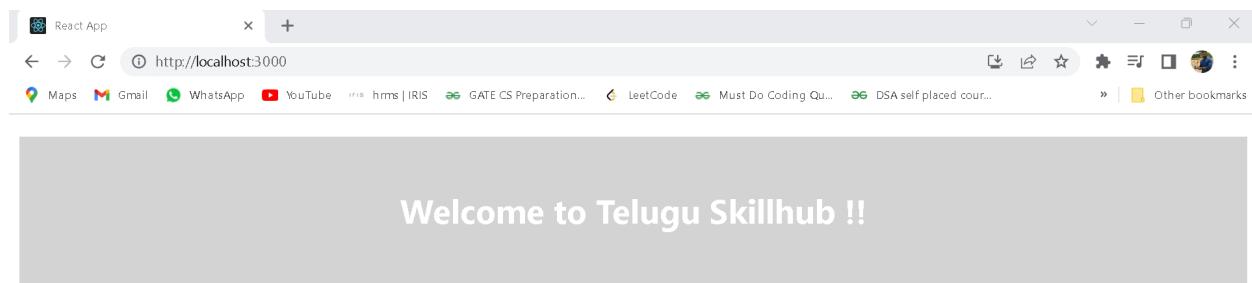
  return (
    <div style= {{margin:"10px"}}>
      <h1 style={styling}>Welcome to Telugu Skillhub !!</h1>
    </div>
  )
}

export default App

```

```
)  
}  
  
export default App
```

O/P:

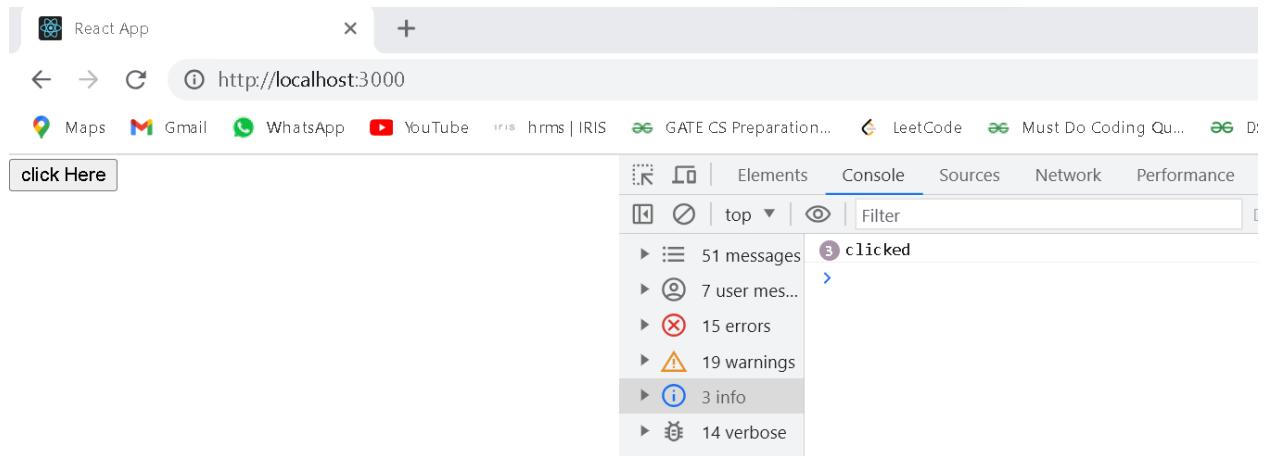


onClick in ReactJs

App.js

```
import React from 'react'  
  
const App = () => {  
  return (  
    <div>  
      <button onClick={() => console.log("clicked")}>click  
Here</button>  
    </div>  
  )  
}  
  
export default App
```

O/P:



How to use useState in ReactJs

App.js

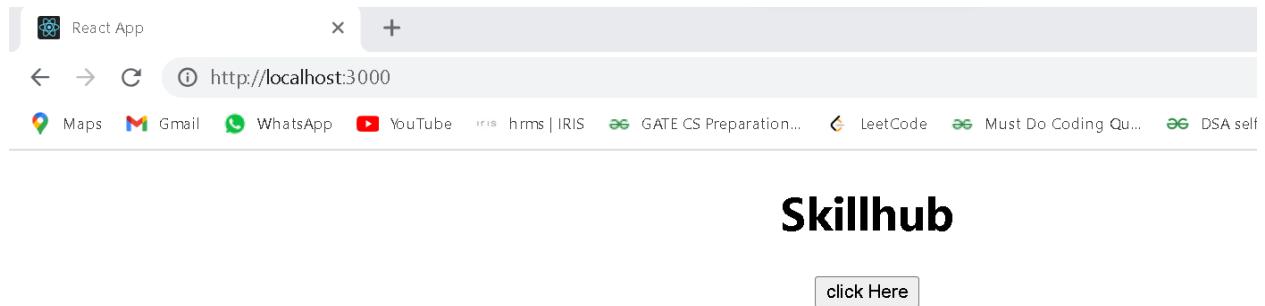
```
import React, { useState } from 'react'

const App = () => {
  const [name, setName] = useState("Skillhub");

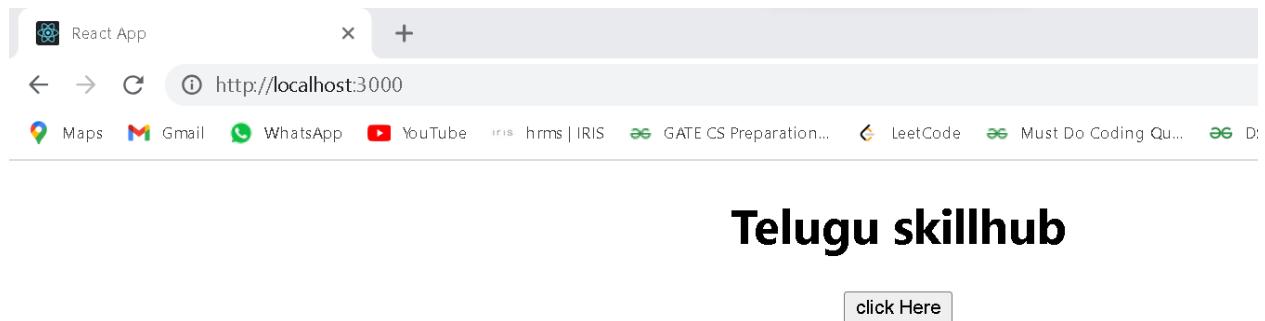
  return (
    <div>
      <center>
        <h1>{ name }</h1>
        <button onClick={() => setName("Telugu skillhub")}>click Here</button>
      </center>
    </div>
  )
}

export default App
```

O/P:



→After Clicking the button it is changed to



App.js

```
import React, { useState } from 'react'

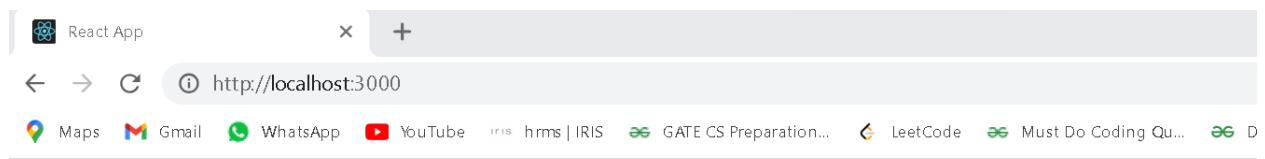
const App = () => {
  const [count, setCount] = useState(0);

  return (
    <div>
      <center>
        <h1>{ count }</h1>
        <button onClick={() => setCount(count+1)}>click
      Here</button>
    
```

```
        </center>
      </div>
    )
}

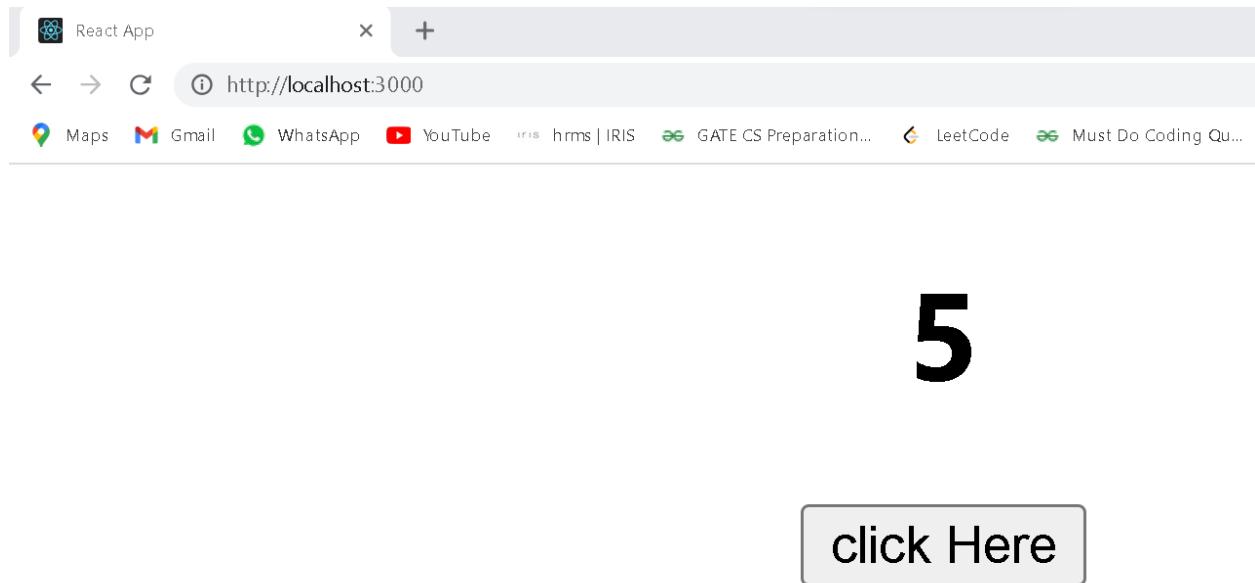
export default App
```

O/P:



click Here

→ Everytime count will change after clicking the button



How to use useEffect in ReactJs

→ **useEffect** will execute **only once** when there are no dependencies immediately after the execution of return.

App.js

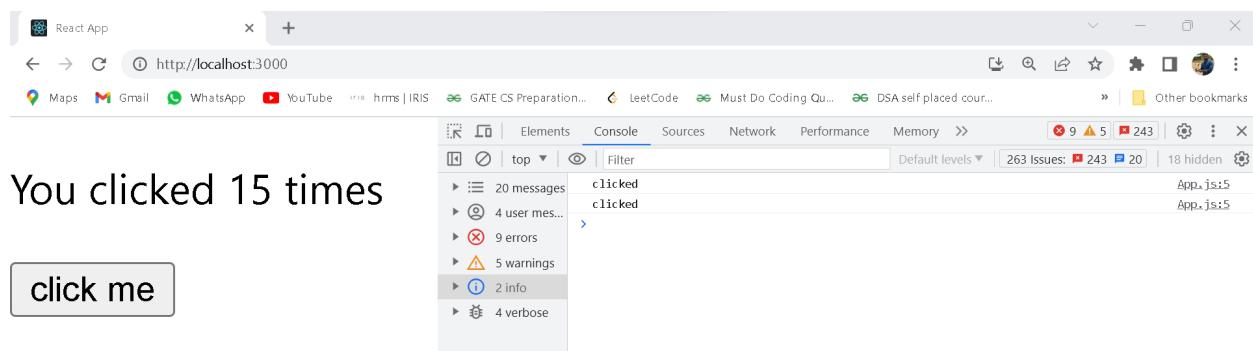
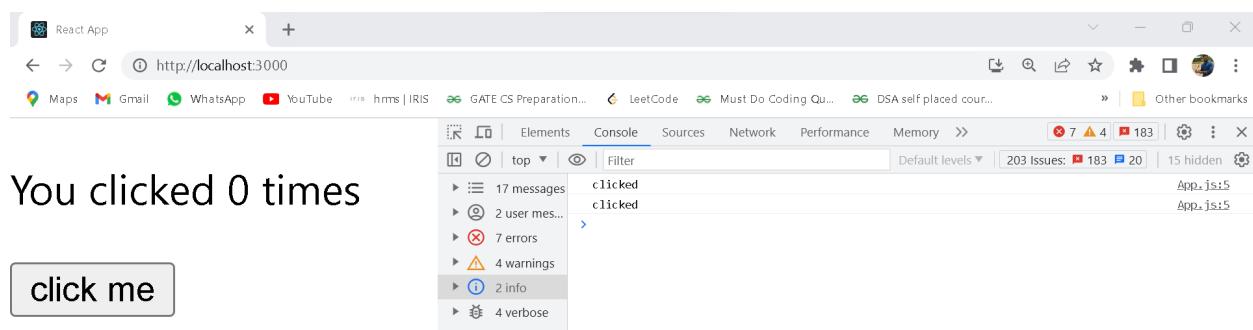
```
import React, { useState, useEffect } from 'react'

const App = () => {
  const [count, setCount] = useState(0);
  useEffect(()=> console.log("clicked"), [])
  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count+1)}>
```

```
    click me
  </button>
</div>
);
}

export default App
```

O/P:



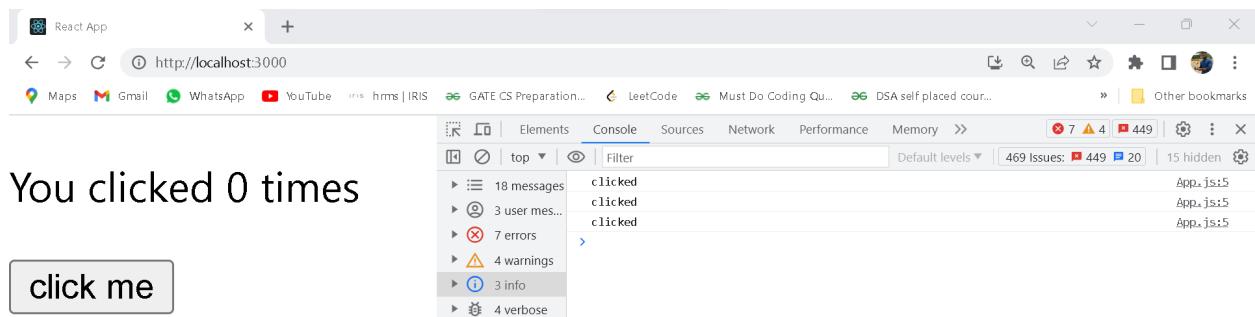
App.js

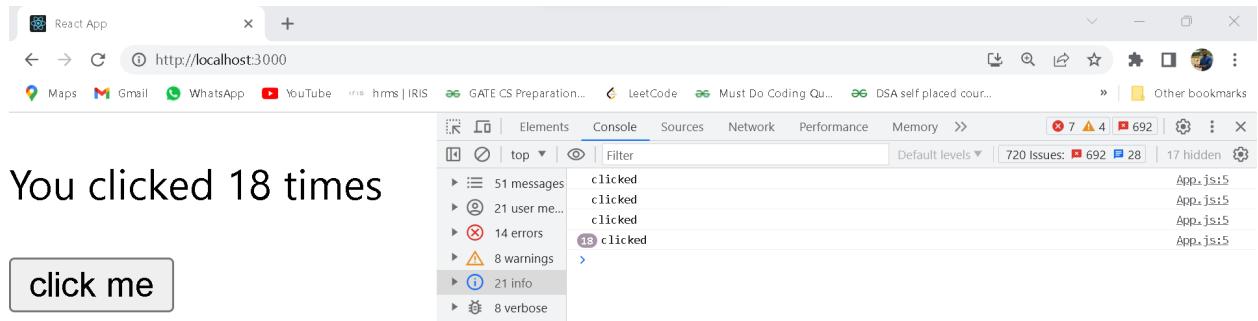
```
import React, { useState, useEffect } from 'react'

const App = () => {
  const [count, setCount] = useState(0);
  useEffect(()=> console.log("clicked"), [count])
  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count+1)}>
        click me
      </button>
    </div>
  );
}

export default App
```

O/P:





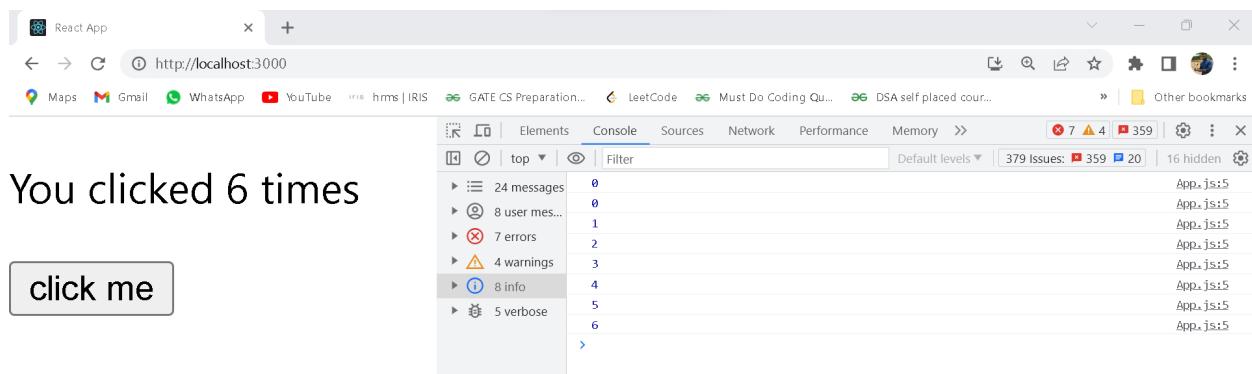
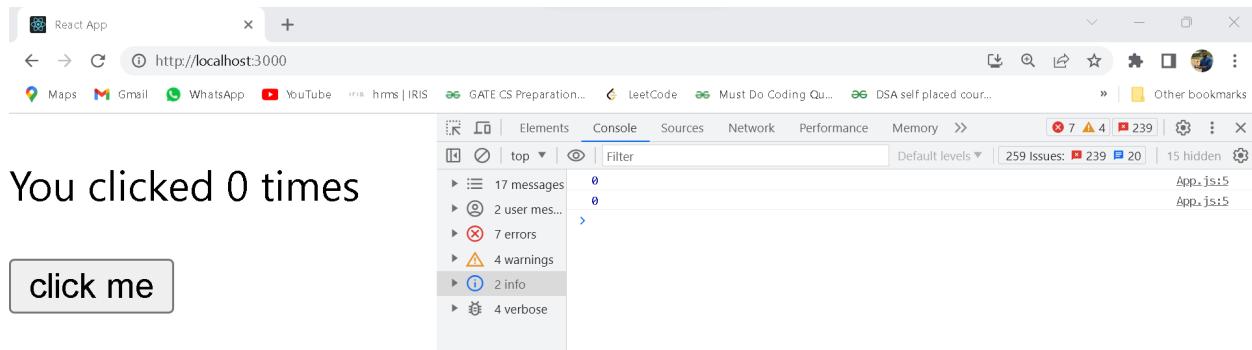
App.js

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

const App = () => {
  const [count, setCount] = useState(0);
  useEffect(()=> console.log(count), [count])
  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count+1)}>
        click me
      </button>
    </div>
  );
}

export default App
```

O/P:



onChange in ReactJs

In React, the `onChange` event handler is commonly used to handle changes to input fields or form elements. It is triggered whenever the value of the input field or form element changes.

App.js

```
import React, {useState} from 'react'

const App = () => {
  const [user, setUser] = useState("")
```

```

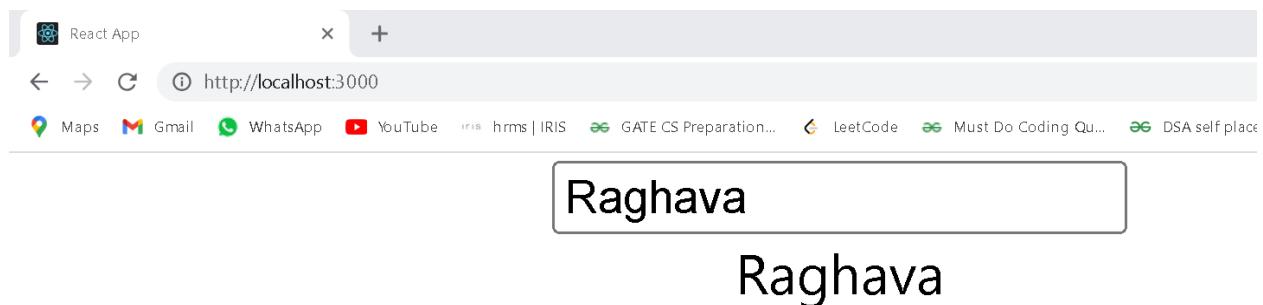
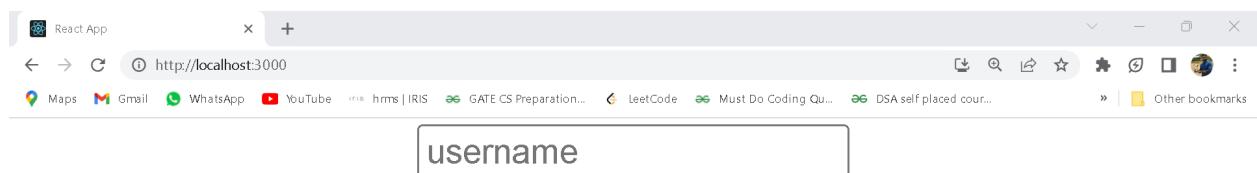
const handle = e =>{
    setUser(e.target.value);
}

return (
    <div>
        <center>
            <input type='text' placeholder='username' value={user}
name='user'
                onChange={handle}
            ></input> <br></br>
            {user}
        </center>
    </div>
)
}

export default App

```

O/P:



onSubmit in ReactJs

→ I have taken data i need to submit in backend we will use onSubmit handler

App.js

```
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(); // Any default event handler it will
manage
    console.log(data);
    //axios.post('', data);    //in that database it will store
  }

  return (
    <div>
      <center>
        <form onSubmit={submitHandler}>
          <input type='text' name='username' value={username}>
        </form>
      </center>
    </div>
  )
}

export default App
```

```

        onChange={onChange} placeholder='Username'></input>
<br></br>

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

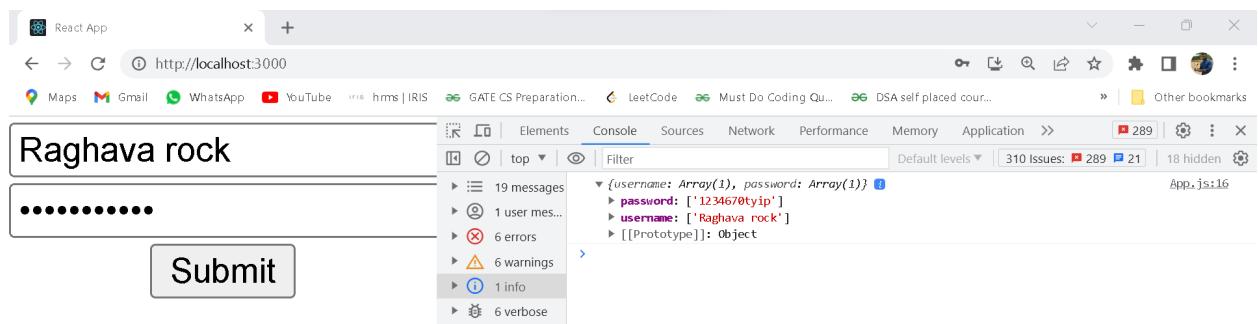
        <input type="submit" name="submit"></input>
    </form>
</center>
</div>
)

}

export default App

```

O/P:



How to use Map Function in ReactJs

In React, the `map()` function is commonly used to iterate over an array and generate a new array of React components or elements based on the original array. It allows you to dynamically render a list of elements based on the data in your application.

App.js

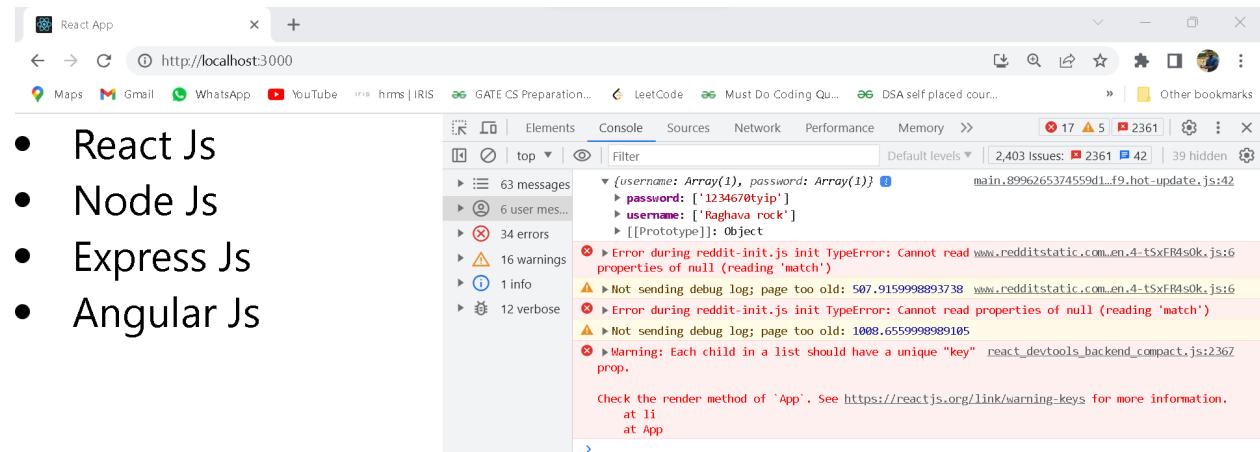
```
import React from 'react'

function App() {
  const arr = ["React Js", "Node Js", "Express Js", "Angular Js"]
  return (
    <div>
      {
        arr.map(
          (value, index) => <li>{ value }</li>
        )
      }
    </div>
  )
}

export default App
```

O/P:

→Error



App.js

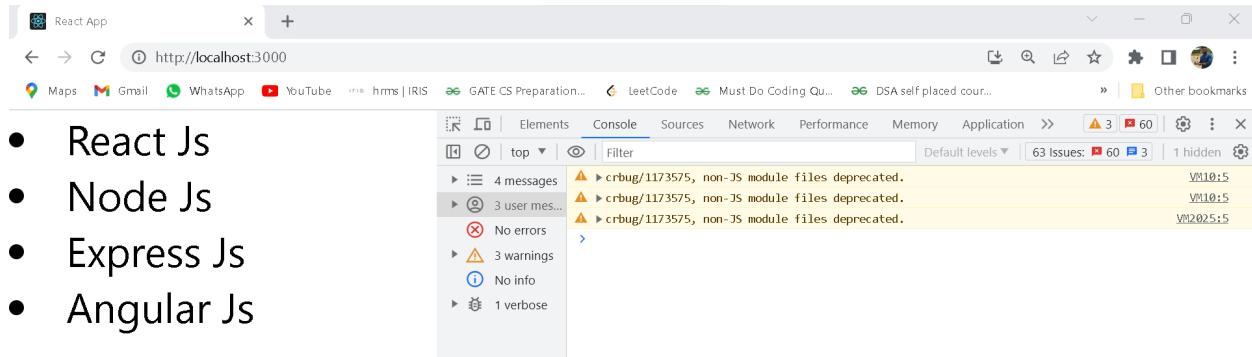
```
import React from 'react'

function App() {
  const arr = ["React Js", "Node Js", "Express Js", "Angular Js"]
  return (
    <div>
      {
        arr.map(
          (value, index) => <li key={index}>{ value }</li>
        )
      }
    </div>
  )
}

export default App
```

O/P:

→**No Error**



App.js

```
import React from 'react'

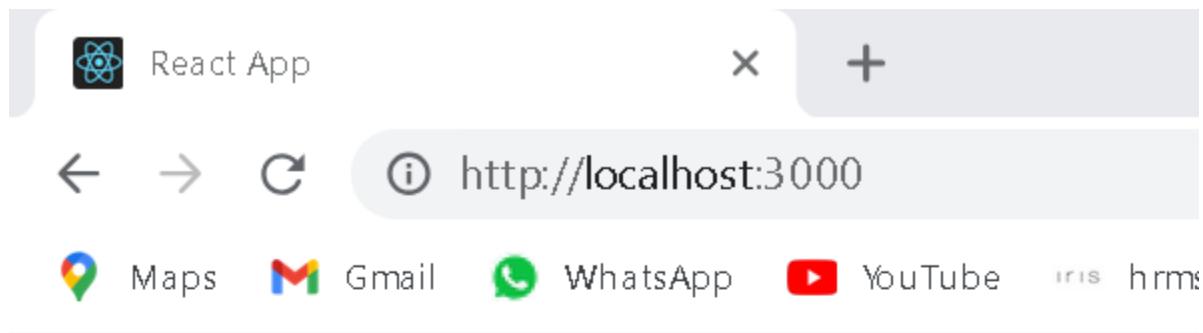
function App() {
  const arr = [
    {
      id:1,
      title:"React Js",
    },
    {
      id:2,
      title:"Node Js",
    },
    {
      id:3,
      title:"Express Js",
    },
    {
      id:4,
      title:"Angular Js"
    }
  ]
  return (
    <ul>
      {arr.map(item=>
        <li key={item.id}>
          {item.title}
        </li>
      )}
    </ul>
  )
}

export default App
```

```
        }
    ]
    return (
        <div>
        {
            arr.map(
                (value)=> <li key={value.id}>{ value.title }</li>
            )
        }
        </div>
    )
}

export default App
```

O/P:



- React Js
- Node Js
- Express Js
- Angular Js

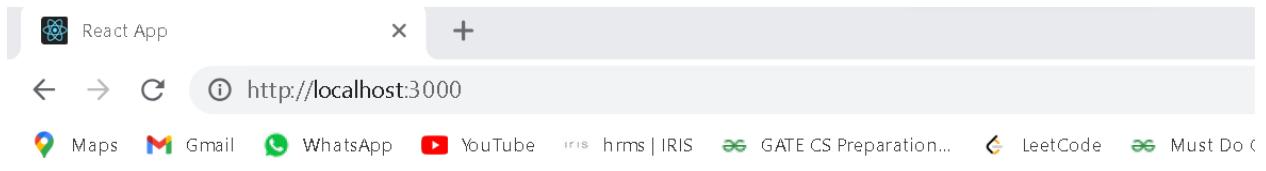
How to use Filter Function in ReactJs

```
array.filter(  
  ( Variable ) => condition..  
)
```

App.js

```
import React from 'react'  
  
function App() {  
  const names = ['James', 'John', 'Raghava', 'Ringo',  
'George'];  
  const filtered = names.filter(  
    (name) => name.includes('J'))  
  return (  
    <div>  
      {  
        filtered.map(item => <li>{item}</li>)  
      }  
    </div>  
  )  
}  
  
export default App
```

O/P:



- James

- John

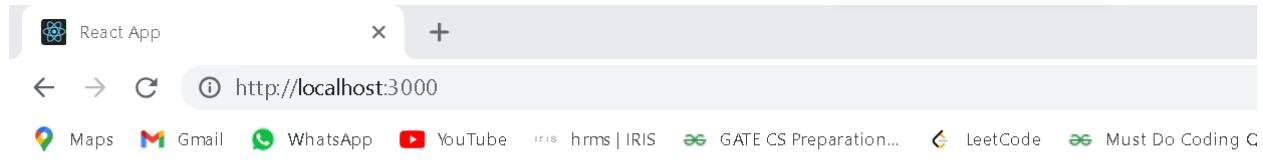
App.js

```
import React from 'react'

function App() {
  const arr = [10,20,30,40,50,60,70];
  const filtered = arr.filter(
    (item) => item > 40
  )
  return (
    <div>
      {
        filtered.map(item => <li>{item}</li>)
      }
    </div>
  )
}

export default App
```

O/P:



- 50
- 60
- 70

Difference Between Export and Export Default

export --> import {Component}

export default --> import Component

→Using Export Default

App.js

```
import React from 'react'
import Header from './components/Header'
import Home from './components/Home'
import Footer from './components/Footer'

function App() {

  return (
    <div>
      <Header />
      <Home />
      <Footer />
    </div>
  )
}


```

```
export default App
```

Home.js

```
import React from 'react'

const Home = () => {
  return (
    <div>
      <h2>Home</h2>
    </div>
  )
}

export default Home
```

Header.js

```
import React from 'react'

const Header = () => {
  return (
    <div>
      <h2>Header</h2>
    </div>
  )
}

export default Header
```

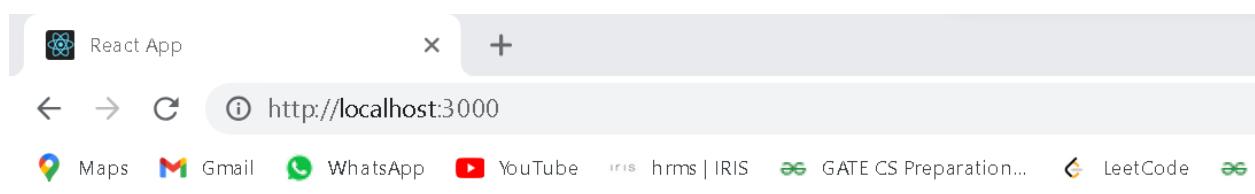
Footer.js

```
import React from 'react'
```

```
const Footer = () => {
  return (
    <div>
      <h2>Footer</h2>
    </div>
  )
}

export default Footer
```

O/P:



Header

Home

Footer

→Using the export only

App.js

```
import React from 'react'
import {Header} from './components/Header'
import {Home} from './components/Home'
import {Footer} from './components/Footer'

function App() {

  return (
    <div>
      <Header />
      <Home />
      <Footer />
    </div>
  )
}

export default App
```

Home.js

```
import React from 'react'

export const Home = () => {
  return (
    <div>
      <h2>Home</h2>
    </div>
  )
}
```

Header.js

```
import React from 'react'

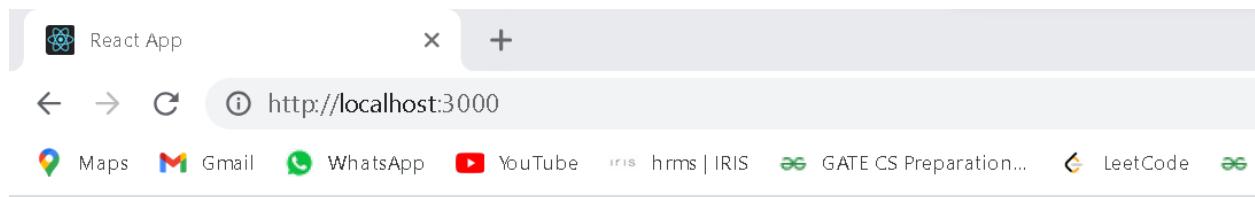
export const Header = () => {
  return (
    <div>
      <h2>Header</h2>
    </div>
  )
}
```

Footer.js

```
import React from 'react'

export const Footer = () => {
  return (
    <div>
      <h2>Footer</h2>
    </div>
  )
}
```

O/P:



Header

Home

Footer

How to Create Login Form With React

Prerequisite concepts

onClick, onChange & onSubmit

useState () - React Hook

Object destructuring

App.js

```
import React, {useState} from 'react'

const App = () => {
  const [data, setData] = useState({
    username : '',
    password : ''
  })

  const {username, password} = data; //Destructuring

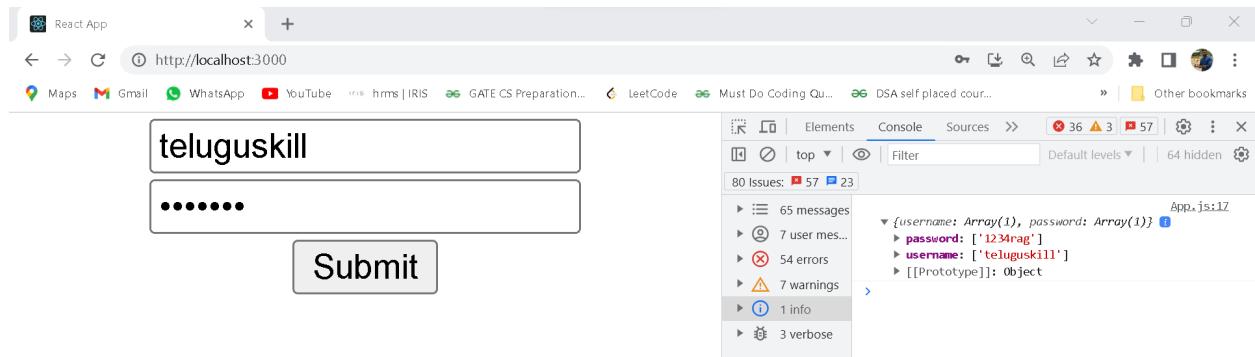
  const changehandler = e =>{
    setData({...data, [e.target.name]:[e.target.value]});
  }
}
```

```
const submitHandler = e =>{
    e.preventDefault(); //If any action is pending it will prevent
    console.log(data);
}

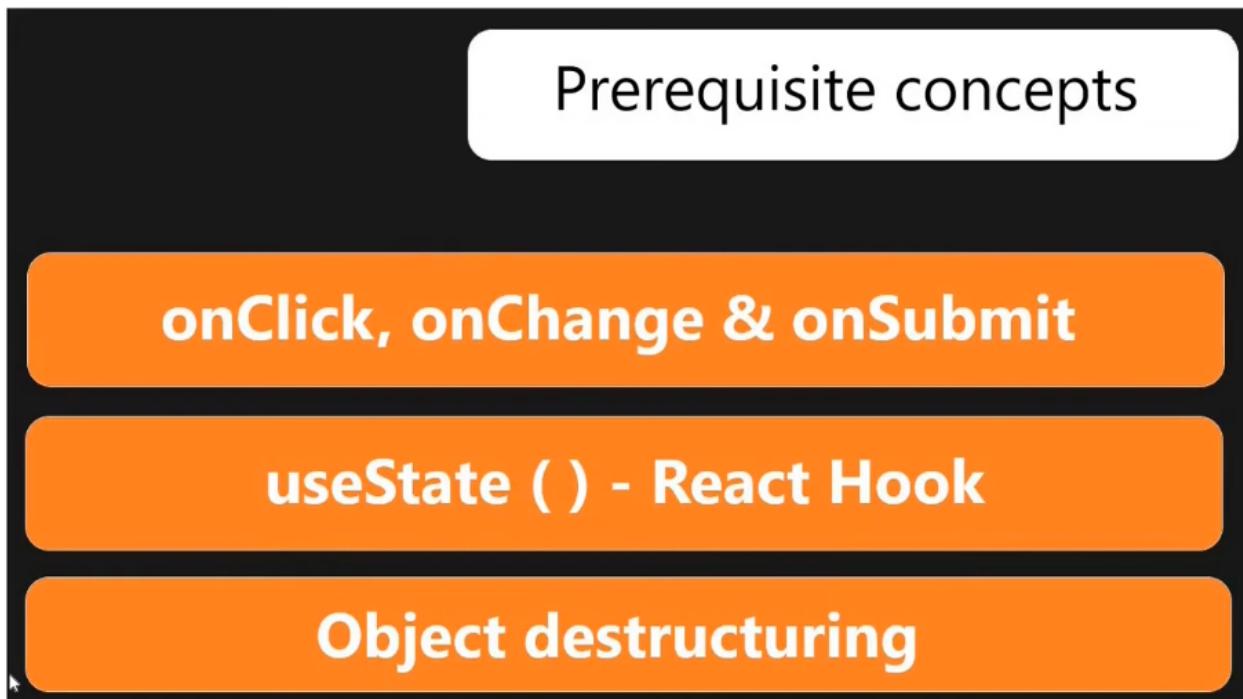
return (
<div>
    <center>
        <form onSubmit={submitHandler}>
            <input type='text' name="username" value={username} onChange={changehandler}></input> <br><br>
            <input type='password' name="password" value={password} onChange={changehandler}></input> <br><br>
            <input type="submit" name="submit"></input>
        </form>
    </center>
</div>
)
}

export default App
```

O/P:



How to Create Sign up Form With React



App.js

```
import React, {useState} from 'react'

const App = () => {
  const [data, setData] = useState({
    username: "",
    email: "",
    password: "",
    confirmPassword:""
  })

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

  const changehandler = e =>{
    setData({...data, [e.target.name]:e.target.value});
  }

  const submitHandler = e =>{
    e.preventDefault(); //If any action is pending it will prevent
    if(password === confirmPassword){
      console.log(data);
    }
    else{
      console.log('Passwords are not matching');
    }
  }

  return (
    <div>
```

```

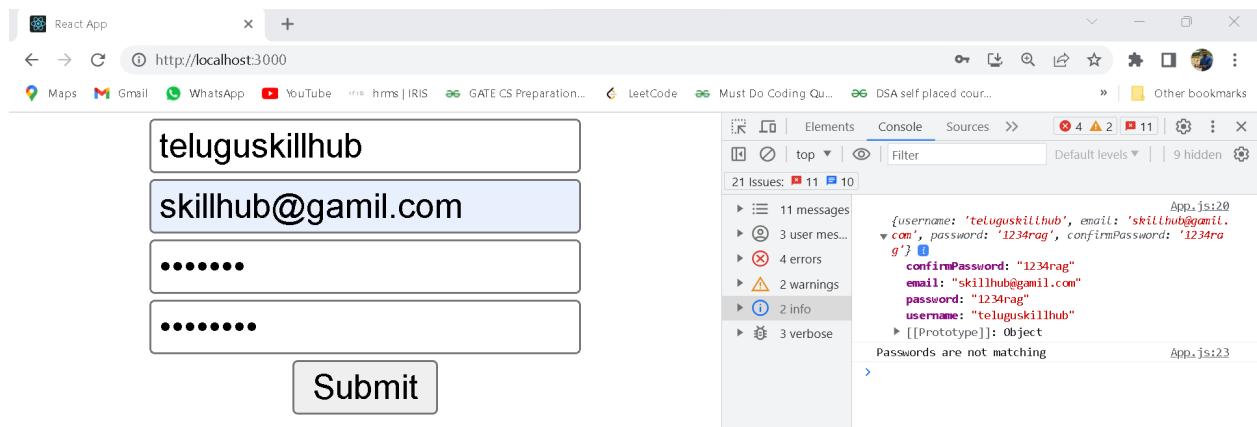
<center>
  <form onSubmit={submitHandler}>
    <input type='text' name="username" value={username}
    onChange={changehandler}></input> <br></br>
    <input type='email' name='email' value={email}
    onChange={changehandler}></input> <br></br>
    <input type='password' name="password"
    value={password} onChange={changehandler}></input> <br></br>
    <input type='password' name="confirmPassword"
    value={confirmPassword} onChange={changehandler}></input>
  <br><br>
    <input type="submit" name="submit"></input>
  </form>
</center>

</div>
)
}

export default App

```

O/P:



Form Validation in ReactJs

In React, the `autoComplete` attribute with the value `off` is used in HTML form elements to disable or turn off the autocomplete feature provided by browsers.

When `autoComplete` is set to `off`, it tells the browser not to display any previously entered values or suggestions for that specific form field. This can be useful in scenarios where you don't want the browser to suggest or autofill values based on user input history.

App.js

```
import React, {useState} from 'react'

const App = () => {
  const [data, setData] = useState({
    username: "",
    email: "",
    password: "",
    confirmPassword: ""
  })

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

  const changehandler = e =>{
    setData({...data, [e.target.name]:e.target.value});
  }

  const submitHandler = e =>{
    e.preventDefault(); //If any action is pending it will prevent
    if(username.length <= 5) {
```

```
        alert("username must be atleast 5 characters");
    }

    else if(password !== confirmPassword) {
        alert("Passwords are not matching");
    }

    else{
        console.log(data);
    }

}

return (
<div>
    <center>
        <form autoComplete='off' onSubmit={submitHandler}>
            <input type='text' name="username" value={username}
onChange={changehandler} placeholder='Username'></input>
<br><br>
            <input type='email' name='email' value={email}
onChange={changehandler} placeholder='Email'></input>
<br><br>
            <input type='password' name="password"
value={password} onChange={changehandler}
placeholder='Password'></input> <br><br>
            <input type='password' name="confirmPassword"
value={confirmPassword} onChange={changehandler}
placeholder='Confirm Password'></input> <br><br>
            {password !== confirmPassword? <p
style={{color:"red"}}>Passwords not matching</p> : null}
            <input type="submit" name="submit"></input>
        </form>
    </center>
)
```

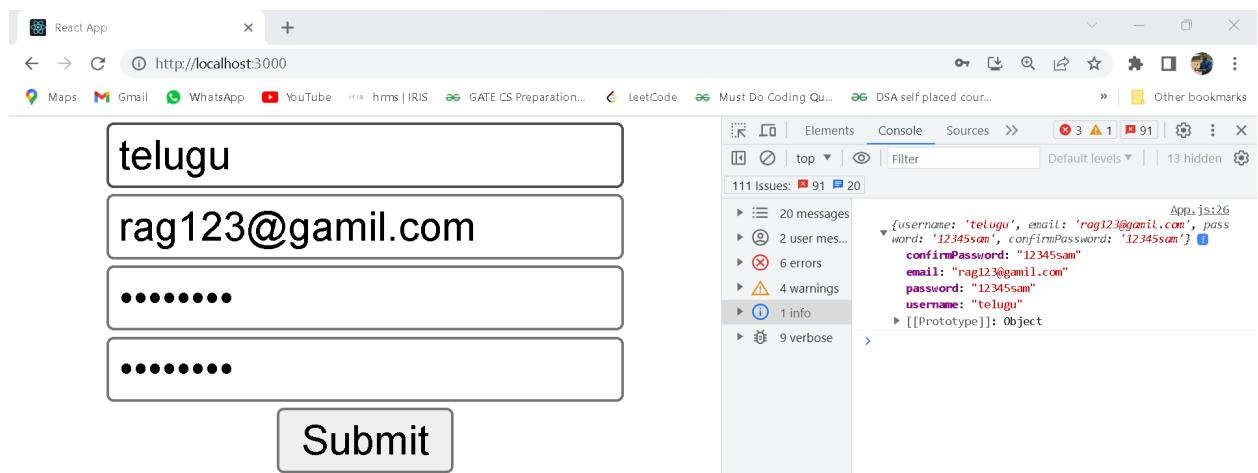
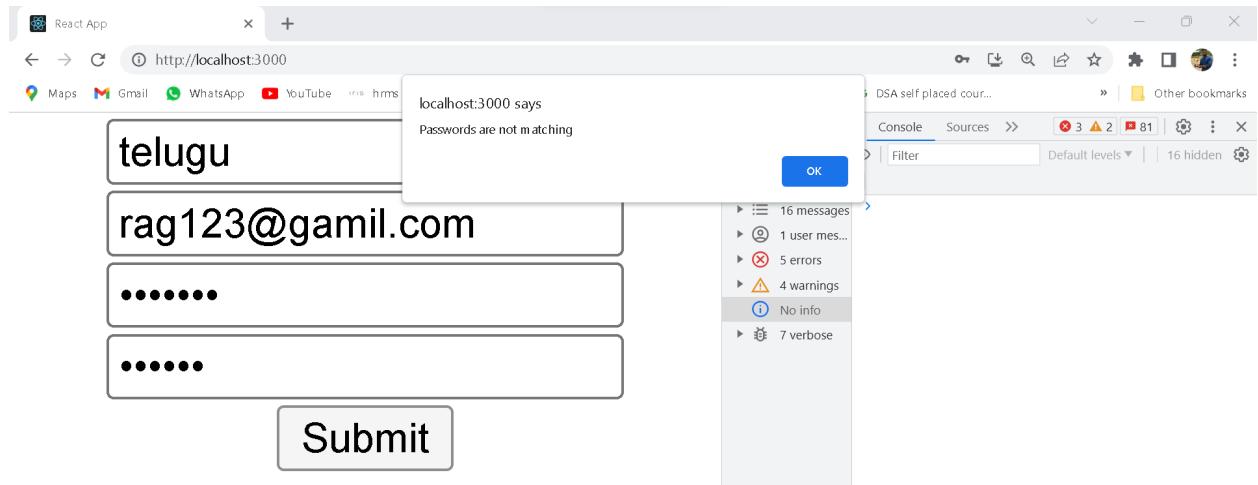
```
</div>
)
}

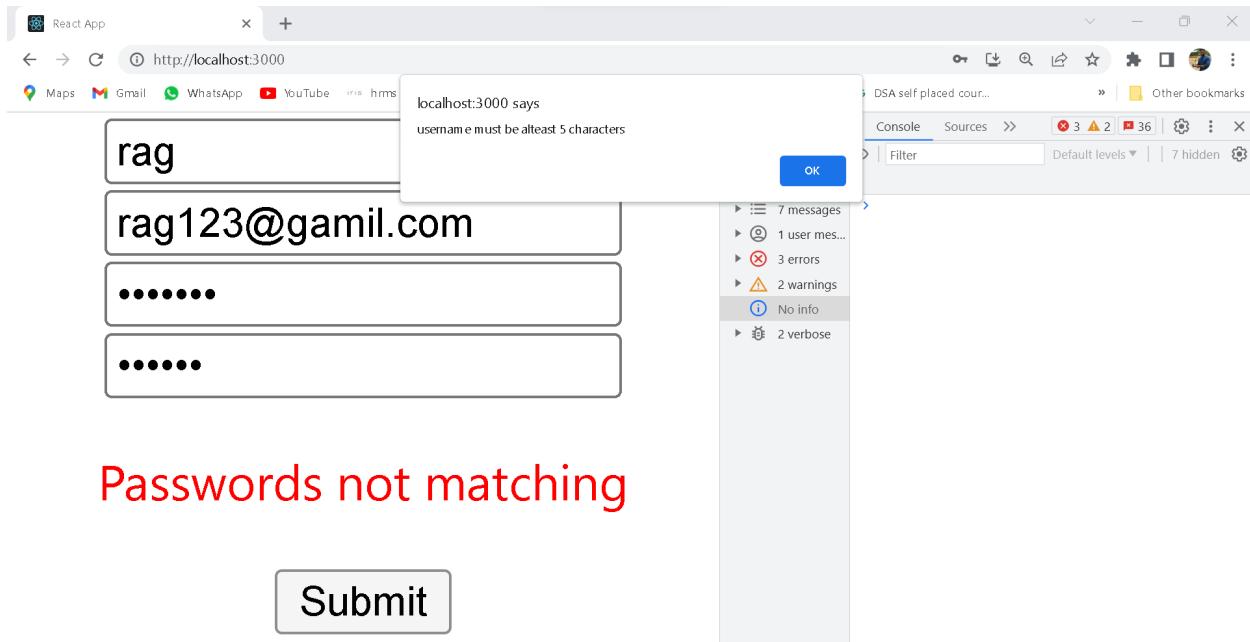
export default App
```

O/P:

A screenshot of a web browser window titled "React App". The address bar shows "http://localhost:3000". The page contains a form with four input fields: "Username", "Email", "Password", and "Confirm Password". Below the inputs is a "Submit" button.

A screenshot of a web browser window titled "React App". The address bar shows "http://localhost:3000". The page contains a form with four input fields: "Username" (containing "rag"), "Email" (containing "rag123@gmail.com"), and two password fields (both containing "....."). An error message box is displayed over the first input field, stating "localhost:3000 says username must be atleast 5 characters". To the right of the form, the developer tools' "Console" tab is open, showing 13 messages, 1 user message, 5 errors, 3 warnings, and 5 verbose entries.

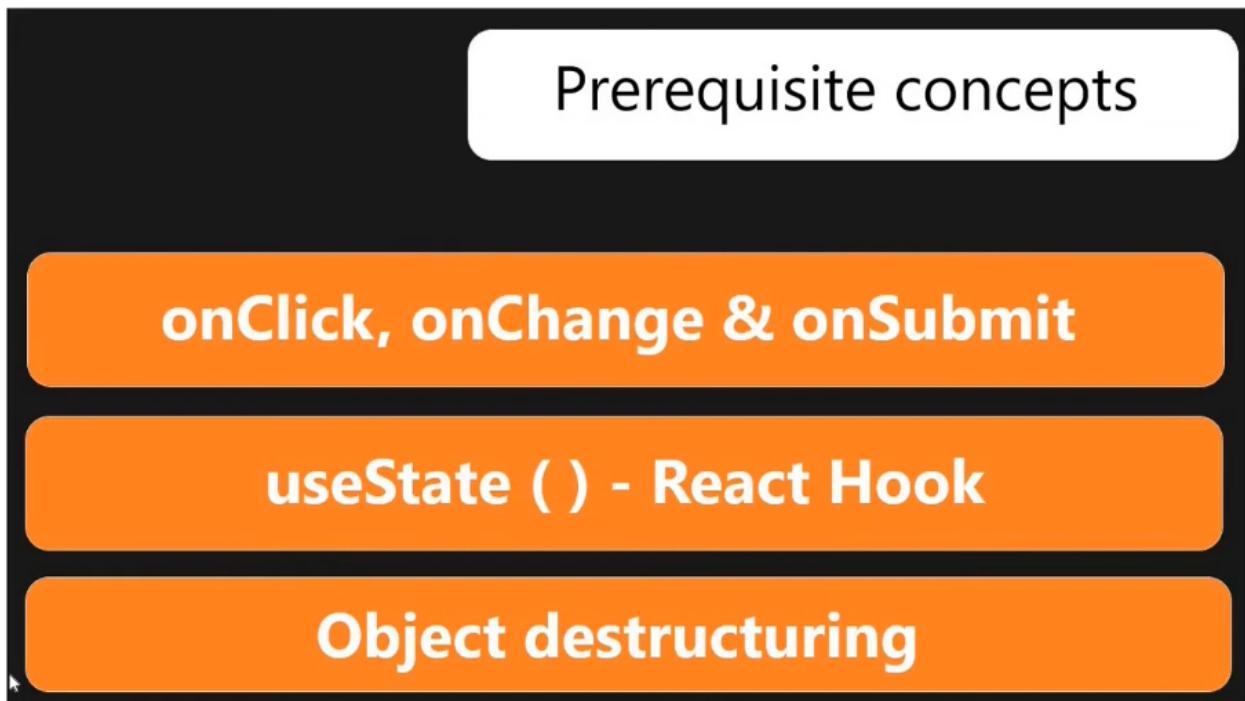




Passwords not matching

Submit

How to Create Calculator React App



App.js

```
import React, {useState} from 'react'
import "./App.css"

const App = () => {
  const [input, setInput] = useState("");
  const [result, setResult] = useState(0);

  const handler = e =>{
    setInput(e.target.value);
  }

  return (
    <div>
      <center>
        <input type='text' value={input} name='input'
onChange={handler}></input><br><br>
        <button onClick={() => setResult(eval(input))}>Result</button>
        <h4>Result is : {result} </h4>

        <button onClick={() =>setInput(input+'1')}>1</button>
        <button onClick={() =>setInput(input+'2')}>2</button>
        <button onClick={() =>setInput(input+'3')}>3</button>
        <button onClick={() =>setInput(input+'4')}>4</button>
        <button onClick={() =>setInput(input+'5')}>5</button>
<br><br>

        <button onClick={() =>setInput(input+'6')}>6</button>
        <button onClick={() =>setInput(input+'7')}>7</button>
        <button onClick={() =>setInput(input+'8')}>8</button>
        <button onClick={() =>setInput(input+'9')}>9</button>
    </center>
  )
}
```

```

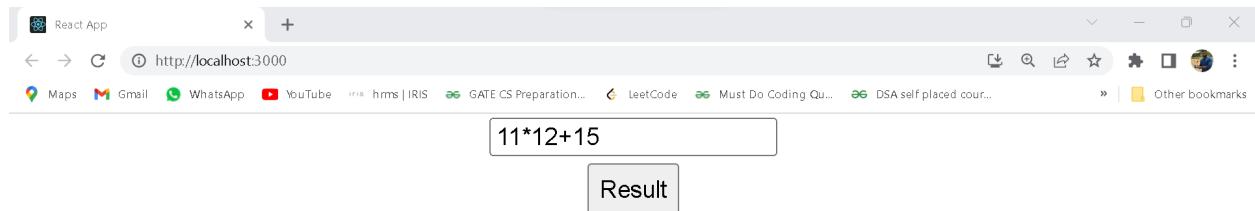
        <button onClick={() =>setInput(input+'0')}>0</button>
<br></br>

        <button onClick={() =>setInput(input+'+')}>+</button>
        <button onClick={() =>setInput(input+'-')}>-</button>
        <button onClick={() =>setInput(input+'*')}>*</button>
        <button onClick={() =>setInput(input+'/')}>/</button>
        <button onClick={() =>setInput('')}>clr</button>
<br></br>
    </center>
</div>
)
}

export default App

```

O/P:



Result is : 147

1	2	3	4	5
6	7	8	9	0
+	-	*	/	clr

How To GET API Data Using Fetch API in ReactJs

In React, you can use the `fetch()` function or the `axios` library to make HTTP requests and fetch data from a server. Both options are commonly used in React applications for handling API calls.

App.js

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

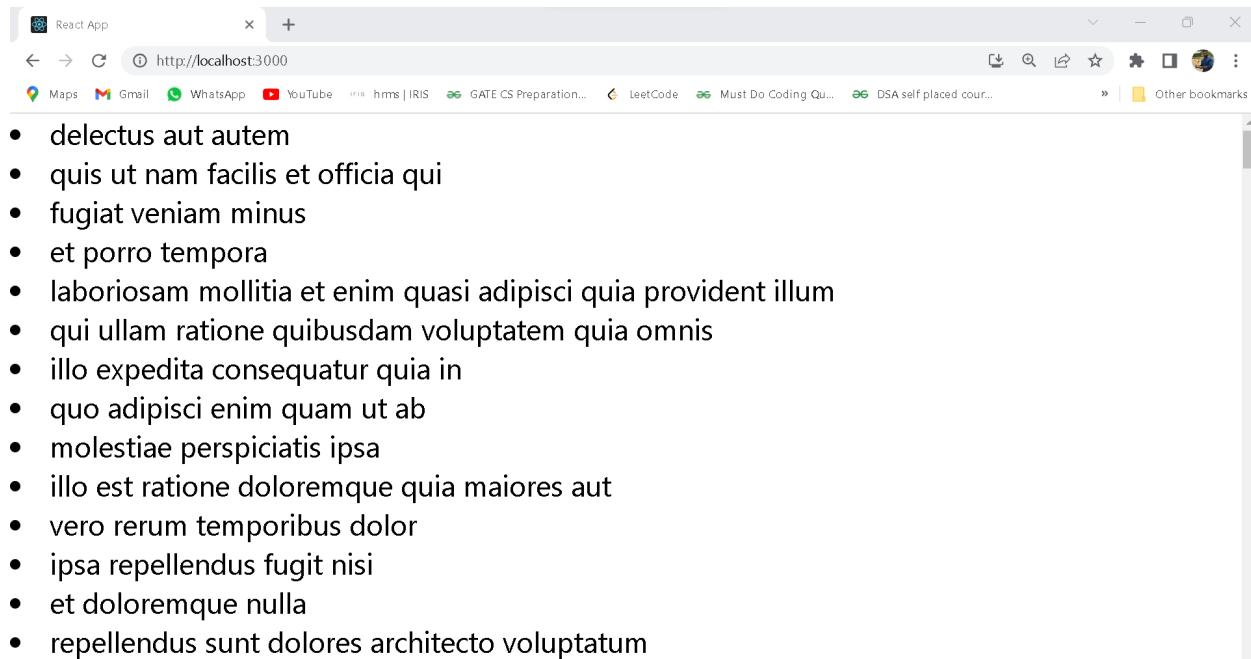
const App = () => {
  const [data, setData] = useState([]);

  useEffect(()=>{
    fetch('https://jsonplaceholder.typicode.com/todos')
    .then(
      response => response.json()
    ) //data is converted into json
    .then(
      json => setData(json)
    ) //if it is converting then we are accessing by using
    json
  ), [])

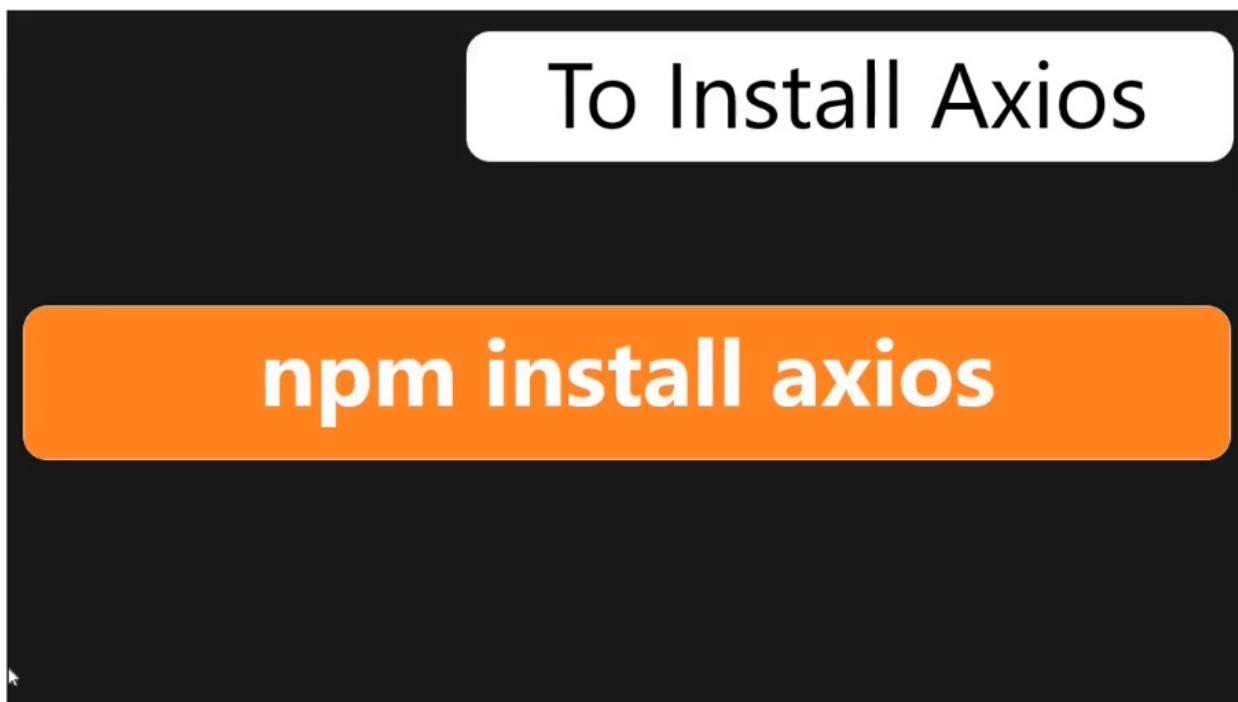
  return (
    <div>
      {data.map(item => <li key={item.id}>{item.title}</li>)}
    </div>
  )
}

export default App
```

O/P:



How To GET API Data Using Axios in ReactJs



App.js

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

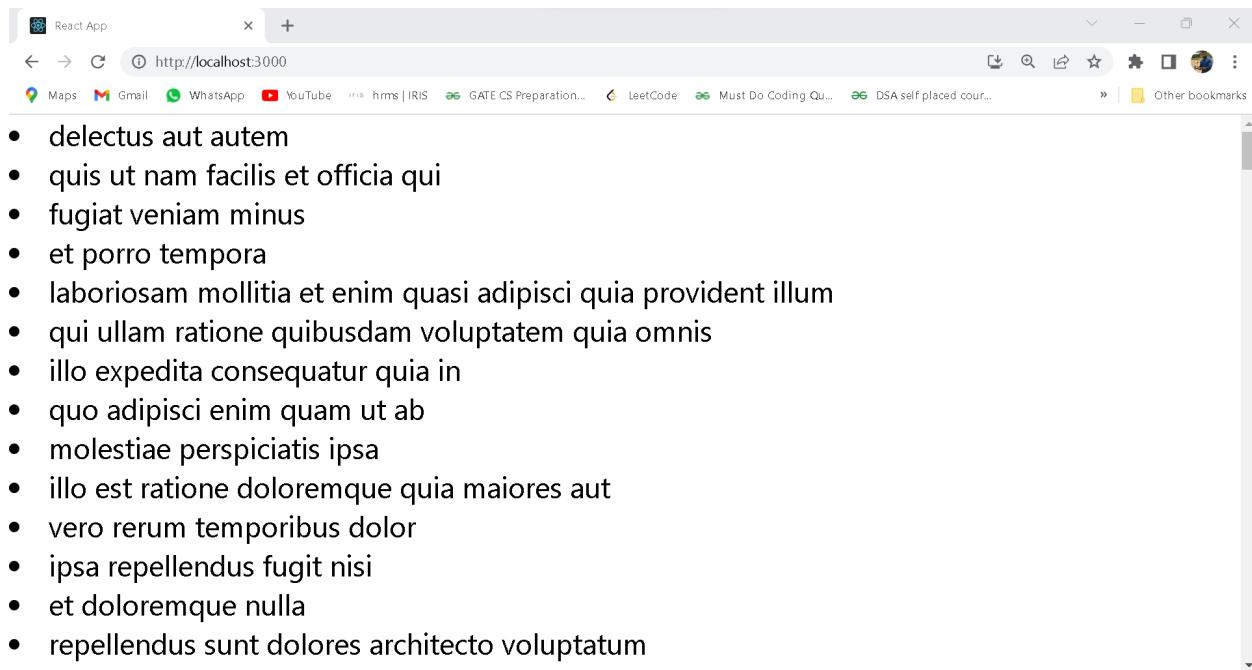
const App = () => {
  const [data, setData] = useState([]);

  useEffect(() =>{
    axios.get('https://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
```

O/P:



How to Use Firebase Realtime Database in ReactJs

Firebase

→<https://firebase.google.com/>

The screenshot shows the official Firebase website. At the top, there's a navigation bar with links to Maps, Gmail, WhatsApp, YouTube, hrmis | IRIS, GATE CS Preparation..., LeetCode, Must Do Coding Qu..., DSA self placed cour..., and Other bookmarks. The main header features the "Firebase" logo with a yellow flame icon, followed by "Products ▾", "Solutions ▾", and "More ▾". A search bar with a magnifying glass icon and the word "Search" is positioned next to it. To the right of the search bar are "Language ▾", "Go to console", and a user profile icon. A dropdown menu is open under "Products" with options: "Build", "Release & Monitor", and "Engage". The main content area has a blue background with white text. It features a large headline: "Make your app the best it can be". Below the headline is a paragraph: "Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world." At the bottom, there are three buttons: "Get started", "Try demo", and "Watch video". To the right of the text, there's a stylized graphic of a hand holding a smartphone, with colorful arrows pointing towards it from different directions.

→Go to Console

The screenshot shows the Firebase homepage. At the top, there's a navigation bar with links to Maps, Gmail, WhatsApp, YouTube, hrms | IRIS, GATE CS Preparation..., LeetCode, Must Do Coding Qu..., and DSA self placed cour... The URL in the address bar is https://console.firebaseio.google.com/u/0/?_gl=1*41zzcc*_ga*NDM5ODg2NzExLjE2ODg4NzYwNTM.*_ga_CW55HF8NVT*MTY4O... Below the navigation is a large yellow logo with the word "Firebase". The main content area features a blue background illustration of two people: one sitting at a desk working on a laptop, and another standing and pointing at a wall covered in floating digital squares. The text "Welcome to Firebase!" is prominently displayed in white. Below it, a subtitle reads "Tools from Google for building app infrastructure, improving app quality, and growing your business". At the bottom left is a white button with the text "Create a project". To its right are links for "View docs" and "Help & Support".

← → C 🔒 https://console.firebaseio.google.com/u/0/?_gl=1*41zzcc*_ga*NDM5ODg2NzExLjE2ODg4NzYwNTM.*_ga_CW55HF8NVT*MTY4O... G 🔍 ☆ ⚙️ ⏴ ⏵ ⏷ ⏸ ⏹ ⏺ ⏻ ⏻ ⏻

Maps Gmail WhatsApp YouTube hrms | IRIS GATE CS Preparation... LeetCode Must Do Coding Qu... DSA self placed cour... Other bookmarks

×

Create a project (Step 1 of 3)

Project name

TeluguSkillhub

teluguskilhub-9115c

I accept the [Firebase terms](#)

I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.

Continue

← → C 🔒 https://console.firebaseio.google.com/u/0/?_gl=1*41zzcc*_ga*NDMSODg2NzExLjE2ODg4NzYwNTM.*_ga_CW55HF8NVT*MTY4O... G 🔍 ☆ ⚙️ ⏴ ⏵ ⏷ ⏸ ⏻ ⏻ ⏻

Maps Gmail WhatsApp YouTube hrms | IRIS GATE CS Preparation... LeetCode Must Do Coding Qu... DSA self placed cour... Other bookmarks

×

Create a project (Step 2 of 2)

A/B testing ⓘ

User segmentation & targeting across Firebase products ⓘ

Crash-free users ⓘ

Event-based Cloud Functions triggers ⓘ

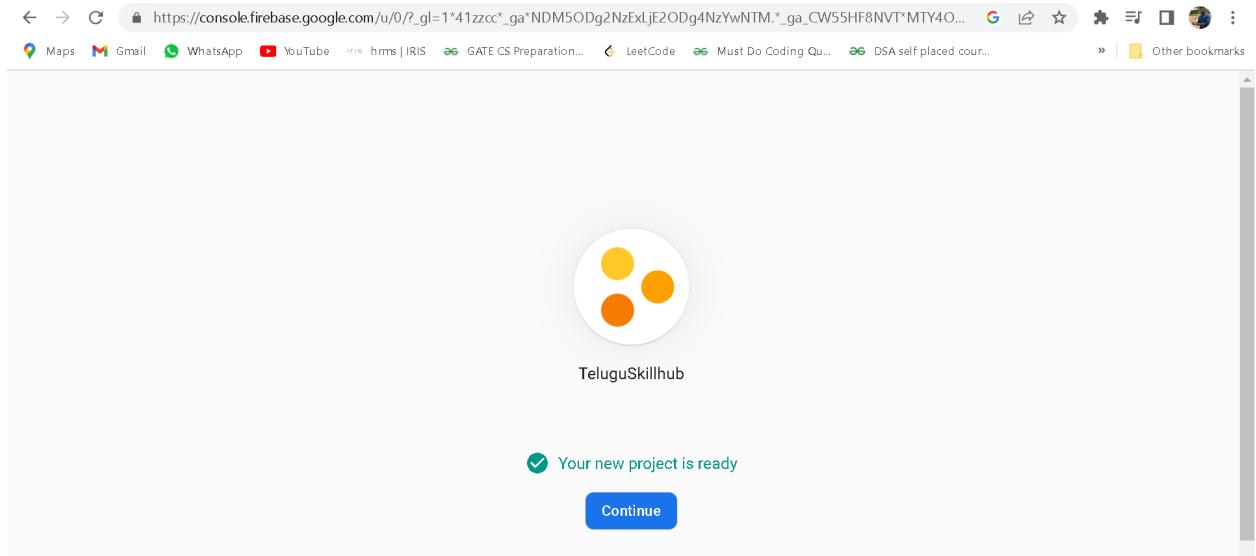
Free unlimited reporting ⓘ

Enable Google Analytics for this project
Recommended

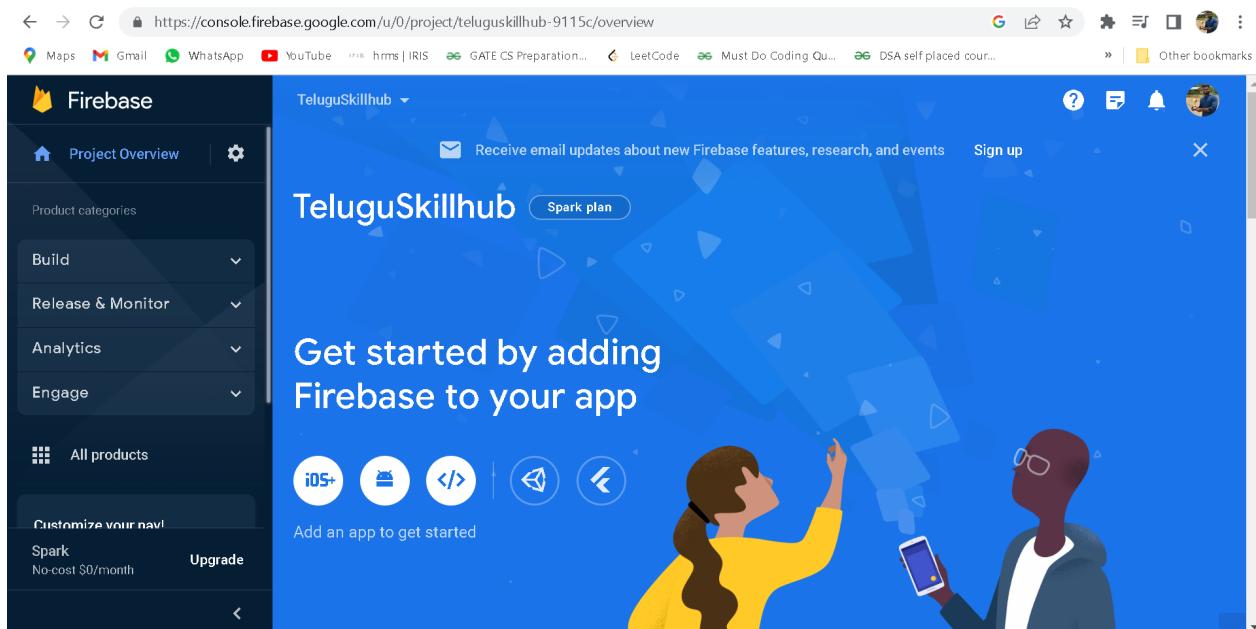
Previous

Create project

→Finally Cloud Platform



→left side menu “Build”click on “Realtime Database”





Project Overview



Product categories

Build



- Authentication
- App Check
- Firestore Database
- Realtime Database
- Extensions
- Storage
- Hosting

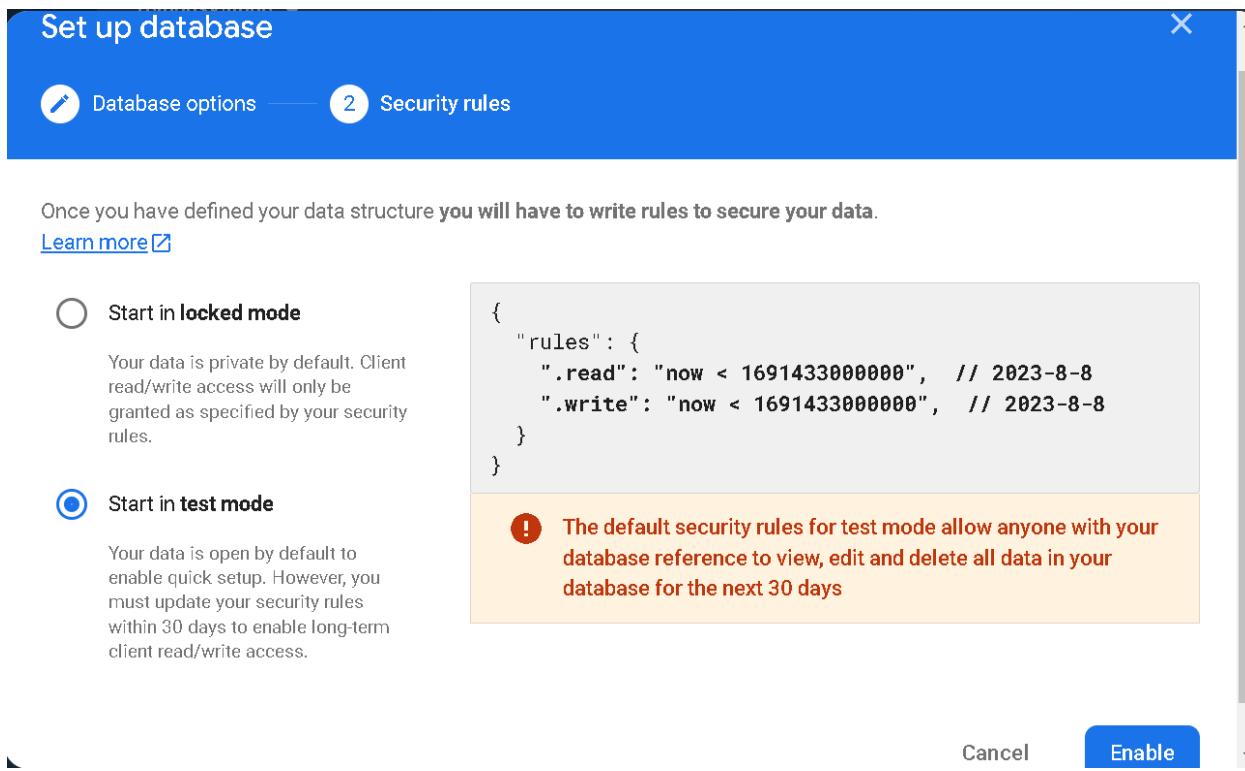
→Create Database

The screenshot shows the Firebase Realtime Database setup page. At the top, there's a navigation bar with links to Maps, Gmail, WhatsApp, YouTube, and other bookmarks. The main title is "Realtime Database" with the subtitle "Store and sync data in real time". A prominent "Create Database" button is centered. On the left, a sidebar titled "Build" lists various services: Authentication, App Check, Firestore Database, Realtime Database (which is selected and highlighted in blue), and Extensions. Below the sidebar, there are "Spark" and "No-cost \$0/month" options. A callout box at the bottom right asks "Is Realtime Database right for you?" with a "Compare Databases" link.

→Click on “Next”

This is a screenshot of the "Set up database" step in the Realtime Database setup wizard. It features a blue header with the title "Set up database" and a progress indicator showing "1 Database options" and "2 Security rules". The main content area displays a note about location settings and a dropdown menu for selecting the "Realtime Database location" set to "United States (us-central1)". At the bottom right, there are "Cancel" and "Next" buttons.

→Select in test mode Click on Enable



Project Overview | [Settings](#)

Realtime Database

Data Rules Backups Usage [Extensions NEW](#)

Protect your Realtime Database resources from abuse, such as billing fraud or phishing [Configure App Check](#)

https://teluguskilhub-9115c-default.firebaseio.com/

https://teluguskilhub-9115c-default.firebaseio.com/:null

Database location: United States (us-central1)

→Rules tab we need to edit

The screenshot shows the 'Edit rules' tab in the Firebase Realtime Database console. At the top, there are tabs for 'Edit rules' (which is selected) and 'Monitor rules'. A blue button labeled 'Rules playground' is located in the top right corner. A prominent message at the top states: 'Default security rules are locked from access' with a star icon. Below this, there is a code editor containing the following JSON:

```
1 /*
2  * Visit https://firebase.google.com/docs/database/security to learn more about security rules.
3  */
4 "rules": {
5     ".read": false,
6     ".write": false
7 }
```

→Initially both read & write are false make it as “true” & Click on Publish

The screenshot shows the 'Edit rules' tab in the Firebase Realtime Database console. At the top, there are tabs for 'Edit rules' (selected) and 'Monitor rules'. Below the tabs, there are buttons for 'unpublished changes', 'Publish', and 'Discard'. A blue button labeled 'Rules playground' is in the top right. A message at the top says: 'Default security rules are locked from access' with a star icon. Below this, there is a code editor with the following JSON, where the '.read' and '.write' fields have been changed to 'true':

```
1 /*
2  * Visit https://firebase.google.com/docs/database/security to learn more about security rules.
3  */
4 "rules": {
5     ".read": true,
6     ".write": true
7 }
```

TeluguSkillhub ▾ Realtime Database

Edit rules Monitor rules

Rules playground

⚠ Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

Learn more ⓘ Dismiss

```
1  {
2    /* Visit https://firebase.google.com/docs/database/security to learn more about security rules. */
3    "rules": {
4      ".read": true,
5      ".write": true
6    }
7 }
```

→ In Data tab “<https://teluguskillhub-9115c-default-rtdb.firebaseio.com/>”

TeluguSkillhub ▾

Realtime Database

Data Rules Backups Usage Extensions NEW

Protect your Realtime Database resources from abuse, such as billing fraud or phishing Configure App Check X

https://teluguskillhub-9115c-default-rtdb.firebaseio.com

https://teluguskillhub-9115c-default-rtdb.firebaseio.com/:null

Database location: United States (us-central1)

App.js

→ In the Url i add the as “register.json”

```
import React, {useState} from 'react'
import axios from 'axios';

const App = () => {
  const [data, setData] = useState({
    fullname: "",
    email: "",
    password: "",
    confirmPassword: ""
  })

  const {fullname, email, password, confirmPassword} = data;
//Destructuring

  const changehandler = e =>{
    setData({...data, [e.target.name]:e.target.value});
  }

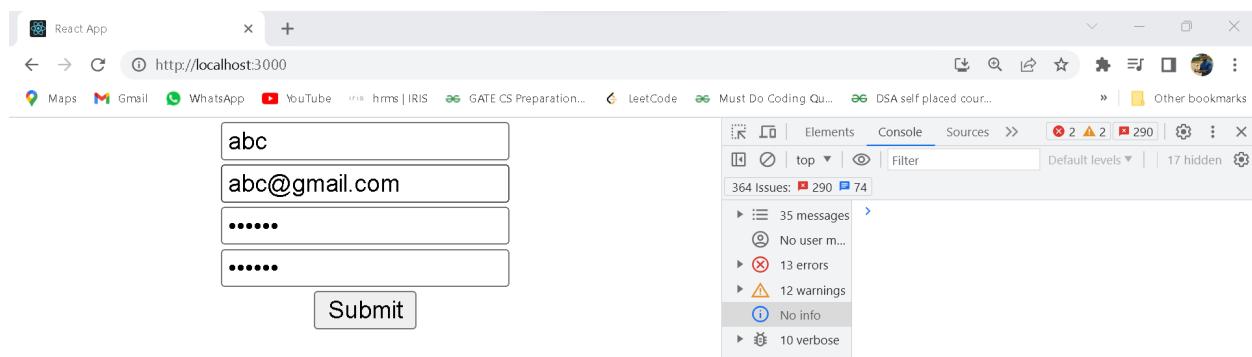
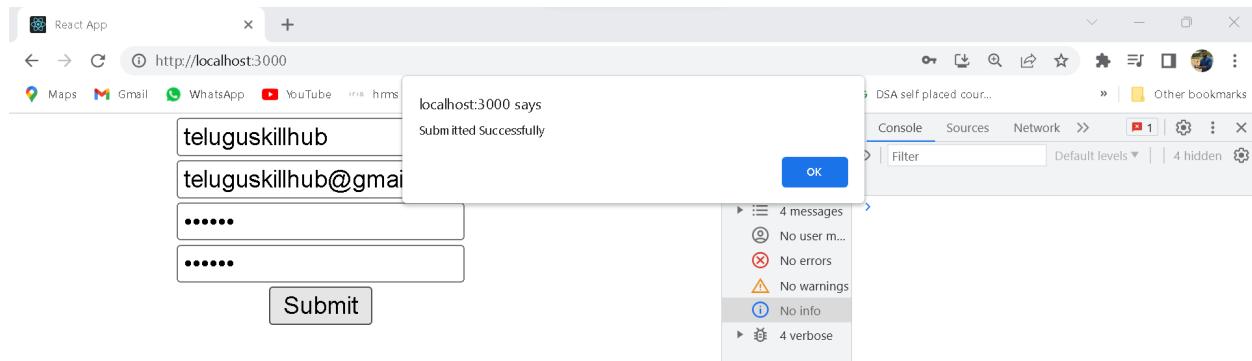
  const submitHandler = e =>{
    e.preventDefault(); //If any action is pending it will prevent
    axios.post('https://teluguskillhub-9115c-default-rtdb.firebaseio.com/register.json', data)
      .then(() =>alert("Submitted Successfully"))
  }
}
```

```
return (
  <div>
    <center>
      <form autoComplete='off' onSubmit={submitHandler}>
        <input type='text' name="fullname" value={fullname}
onChange={changehandler} placeholder='Fullname'></input>
<br></br>
        <input type='email' name='email' value={email}
onChange={changehandler} placeholder='Email'></input>
<br></br>
        <input type='password' name="password"
value={password} onChange={changehandler}
placeholder='Password'></input> <br></br>
        <input type='password' name="confirmPassword"
value={confirmPassword} onChange={changehandler}
placeholder='Confirm Password'></input> <br></br>
        {password !== confirmPassword? <p
style={{color:"red"}}>Passwords not matching</p> : null}
        <input type="submit" name="submit"></input>
      </form>
    </center>

  </div>
)
}

export default App
```

O/P:



A screenshot of the Firebase Realtime Database console at <https://console.firebaseio.google.com/u/0/project/teluguskillhub-9115c/database/teluguskillhub-9115c-default-rtbd/data>. The database structure shows two registered users under the "register" node:

```
https://teluguskillhub-9115c-default-rtbd.firebaseio.com/
  register
    -Nzsy4tukjs8X3VzLtg
      confirmPassword: "123456"
      email: "teluguskillhub@gmail.com"
      fullname: "teluguskillhub"
      password: "123456"
    -NZsyetPJrryP Eg3kUJF
      confirmPassword: "123rag"
      email: "abc@gmail.com"
      fullname: "abc"
      password: "123rag"
```

In React, you can use the axios library to make HTTP requests and handle API calls. axios provides a convenient and widely-used way to perform asynchronous operations, including GET, POST, PUT, DELETE requests, and more.

1. GET Request

```
import axios from 'axios';

axios.get('/api/data')
  .then(response => {
    // Handle the response
    console.log(response.data);
  })
  .catch(error => {
    // Handle the error
    console.error(error);
  });
}
```

The axios.get() method is used to make a GET request to the specified URL (/api/data in this example). The response can be accessed in the .then() block, and any errors are caught in the .catch() block.

2. Post Request

```
import axios from 'axios';

axios.post('/api/data', { name: 'John', age: 30 })
  .then(response => {
    // Handle the response
    console.log(response.data);
  })
  .catch(error => {
    // Handle the error
  });
}
```

```
    console.error(error);
}) ;
```

The `axios.post()` method is used to make a POST request to the specified URL (`/api/data` in this example). The second argument is the data object that will be sent in the request body.

3. Put Request

```
import axios from 'axios';

axios.put('/api/data/123', { name: 'John', age: 35 })
  .then(response => {
    // Handle the response
    console.log(response.data);
  })
  .catch(error => {
    // Handle the error
    console.error(error);
  }) ;
```

The `axios.put()` method is used to make a PUT request to update an existing resource. In this example, the URL `/api/data/123` represents the resource with ID 123, and the second argument is the updated data.

4. Delete Request

```
import axios from 'axios';

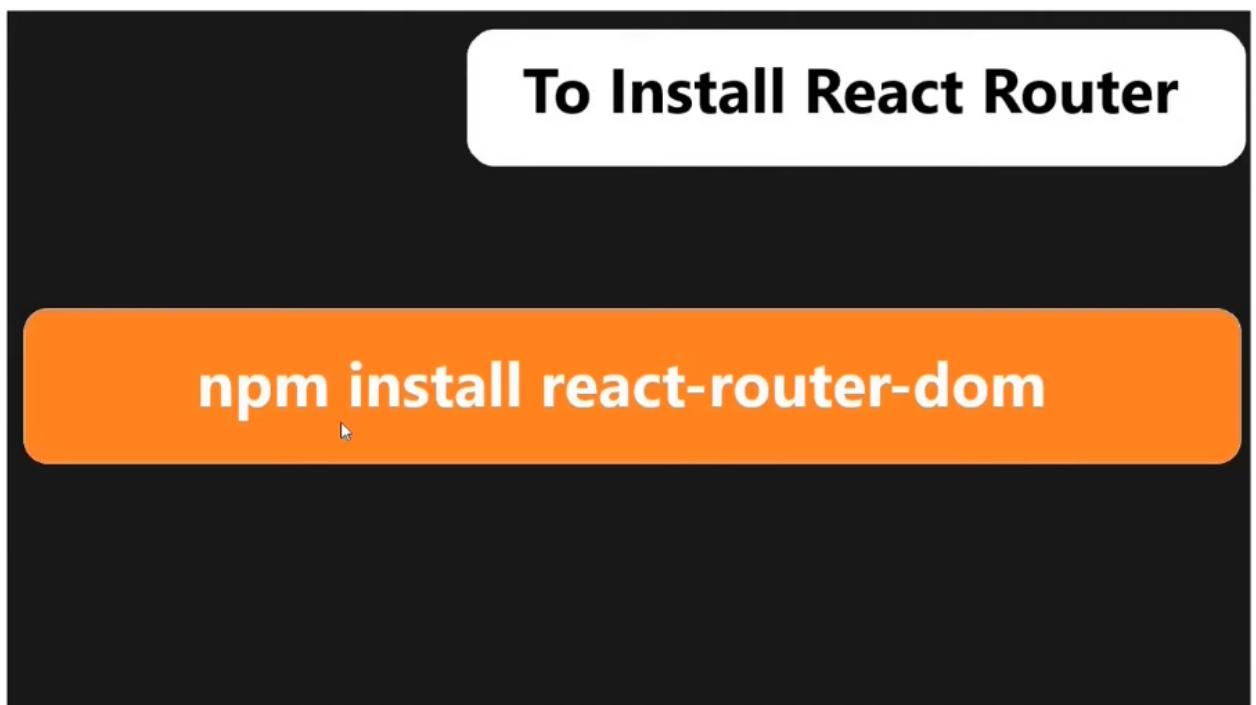
axios.delete('/api/data/123')
  .then(response => {
    // Handle the response
    console.log(response.data);
  })
```

```
.catch(error => {
  // Handle the error
  console.error(error);
}) ;
```

The axios.delete() method is used to make a DELETE request to remove a resource. In this example, the URL /api/data/123 represents the resource with ID 123.

How to use React Router in ReactJs

React Router → It handle multiple Page application



Components To Use

```
import { BrowserRouter, Switch, Route }
```

```
import { Link }
```

Syntax

```
<BrowserRouter>
  <Switch>
    <Route path="/" exact component={Name} />
    .....
    <Route path="/*" exact component={*} />
  </Switch>
</BrowserRouter>
```

App.js

```
import React from 'react';
import Navbar from './Navbar';
import "./App.css";
import {BrowserRouter, Routes, Route} from 'react-router-dom';
import Home from './Home';
import Dashboard from './Dashboard';
import About from './About';

const App = () => {
  return (
    <div>
      <BrowserRouter>
        <Navbar />
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/dashboard" element={<Dashboard />} />
          <Route path="/about" element={<About />} />
        </Routes>
      </BrowserRouter>
    </div>
  );
};

export default App;
```

Navbar.js

```
import React from 'react'
import { Link } from 'react-router-dom'

const Navbar = () => {
  return (
    <div className="navbar">
      <ul className="navbar-links">
        <Link to="/"><li>Home</li></Link>
        <Link to="/dashboard"><li>Dashboard</li></Link>
        <Link to="/about"><li>About</li></Link>
      </ul>
    </div>
  )
}

export default Navbar
```

About.js

```
import React from 'react'

const About = () => {
  return (
    <div>
      <center>
        <h4>Your in About Page</h4>
      </center>
    </div>
  )
}
```

```
}
```

```
export default About
```

Dashboard.js

```
import React from 'react'

const Dashboard = () => {
  return (
    <div>
      <center>
        <h4>Welcome to Dashboard Page.</h4>
      </center>
    </div>
  )
}

export default Dashboard
```

Home.js

```
import React from 'react'

const Home = () => {
  return (
    <div>
      <center>
        <h4>Welcome To Home Page !!</h4>
      </center>
    </div>
  )
}

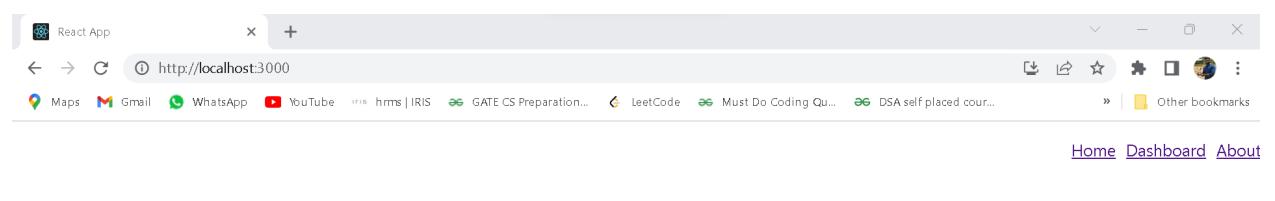
export default Home
```

```
)  
}  
  
export default Home
```

App.css

```
.navbar {  
  display: flex;  
  justify-content: flex-end;  
}  
  
.navbar-links {  
  list-style: none;  
  display: flex;  
}  
  
.navbar-links li {  
  margin-left: 10px;  
}
```

O/P:





Redirect Component in ReactJs

- Redirect function is replaced by **useNavigate**
- This can be used in both class level & Functional level

Home.js

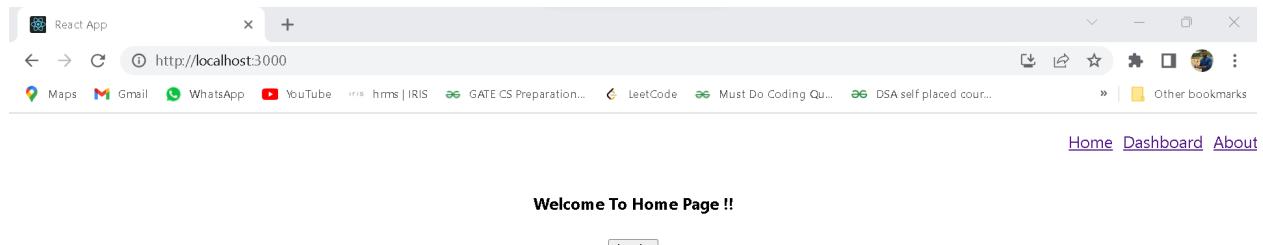
```
import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';

const Home = () => {
  const [auth, setAuth] = useState(false);
  const navigate = useNavigate();

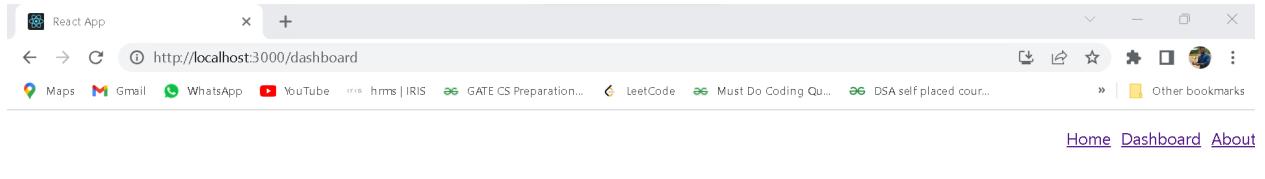
  const handleLogin = () => {
    setAuth(true);
    navigate('/dashboard');
  };
}
```

```
if (auth) {  
    return null; // or you can render a loading state if  
needed  
}  
  
return (  
    <div>  
        <center>  
            <h4>Welcome To Home Page !!</h4>  
            <button onClick={handleLogin}>Login</button>  
        </center>  
    </div>  
);  
};  
  
export default Home;
```

O/P:



→ Click on the Login Page it is Redirect to dashboard page



useHistory hook in ReactJs

→ useHistory can be used only in Function level Component

→ React Router version 6 afterwards there is no “useHistory” we can use only “useNavigate”

Home.js

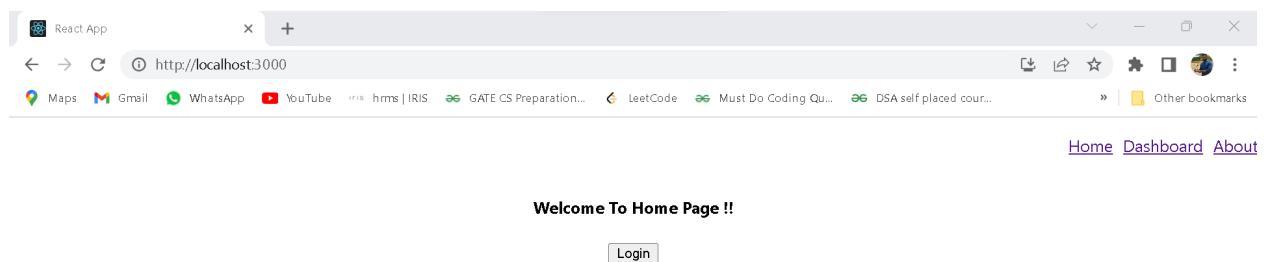
```
import React from 'react';
import { useNavigate } from 'react-router-dom';

const Home = () => {
  const navigate = useNavigate();

  return (
    <div>
      <center>
        <h4>Welcome To Home Page !!</h4>
        <button onClick={() => navigate('/dashboard')}>Login</button>
      </center>
    </div>
  );
};
```

```
export default Home;
```

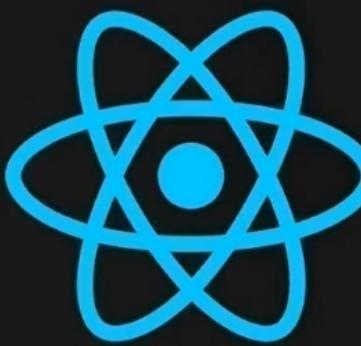
O/P:



→Click on the Login Page it is Redirect to dashboard page



How to Use URL Parameters in ReactJs



Path Params vs Query Params

Path Params

```
<Route path="/profile/:name" component={Profile} />
```

`http://localhost:3000/profile/teluguskillhub`

variable --> name = teluguskillhub

Query Params

```
<Route path="/profile" component={Profile} />
```

http://localhost:3000/profile?name=teluguskillhub

variable ---> name = teluguskillhub

→Path Params

```
import React from 'react';
import { useParams } from 'react-router-dom';

const Home = () => {
  const { username } = useParams();

  return (
    <div>
      <h2>User Profile</h2>
      <p>Username: {username}</p>
    </div>
  );
}

export default Home;
```

→Query Params

```
import React from 'react';
import { useLocation } from 'react-router-dom';

const UserProfile = () => {
  const location = useLocation();
  const queryParams = new URLSearchParams(location.search);
  const username = queryParams.get('username');

  return (
    <div>
      <h2>User Profile</h2>
      <p>Username: {username}</p>
    </div>
  );
}

export default UserProfile;
```

How to Use URL Path Parameters in ReactJs

App.js

```
import React from 'react';
import Navbar from './Navbar';
import "./App.css";
import { BrowserRouter, Routes, Route} from 'react-router-dom';
import Home from './Home';
import Dashboard from './Dashboard';
import About from './About';
```

```
const App = () => {
  return (
    <div>
      <BrowserRouter>
        <Navbar />
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/dashboard/:name" element={<Dashboard />} />
          <Route path="/about" element={<About />} />
        </Routes>
      </BrowserRouter>
    </div>
  );
};

export default App;
```

Dashboard.js

```
import React from 'react';
import { useParams } from 'react-router-dom';

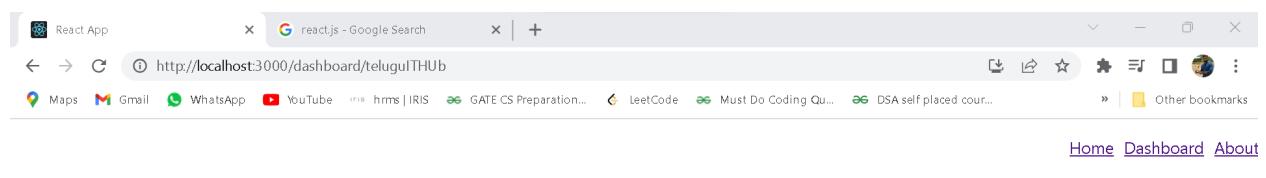
const Dashboard = () => {
  const { name } = useParams();

  return (
    <div>
      <center>
        <h4>Dashboard Profile name is: {name}</h4>
      </center>
    </div>
  );
};

export default Dashboard;
```

```
) ;  
};  
  
export default Dashboard;
```

O/P:



→Query Params

App.js

```
import React from 'react';  
import Navbar from './Navbar';  
import "./App.css";  
import {BrowserRouter, Routes, Route} from 'react-router-dom';  
import Home from './Home';  
import Dashboard from './Dashboard';  
import About from './About';  
  
const App = () => {  
  return (  
    <div>  
      <BrowserRouter>  
        <Navbar />  
        <Routes>  
          <Route path="/" element={<Home />} />  
          <Route path="/dashboard" element={<Dashboard />} />
```

```
        <Route path="/about" element={<About />} />
      </Routes>
    </BrowserRouter>
  </div>
);
};

export default App;
```

Dashboard.js

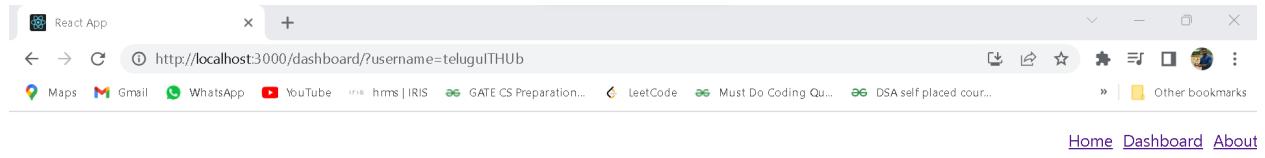
```
import React from 'react';
import { useLocation } from 'react-router-dom';

const UserProfile = () => {
  const location = useLocation();
  const queryParams = new URLSearchParams(location.search);
  const username = queryParams.get('username');

  return (
    <div>
      <h2>User Profile</h2>
      <p>Username: {username}</p>
    </div>
  );
};

export default UserProfile;
```

O/P:



User Profile

Username: telugulTHUB

How to Import Images in ReactJs

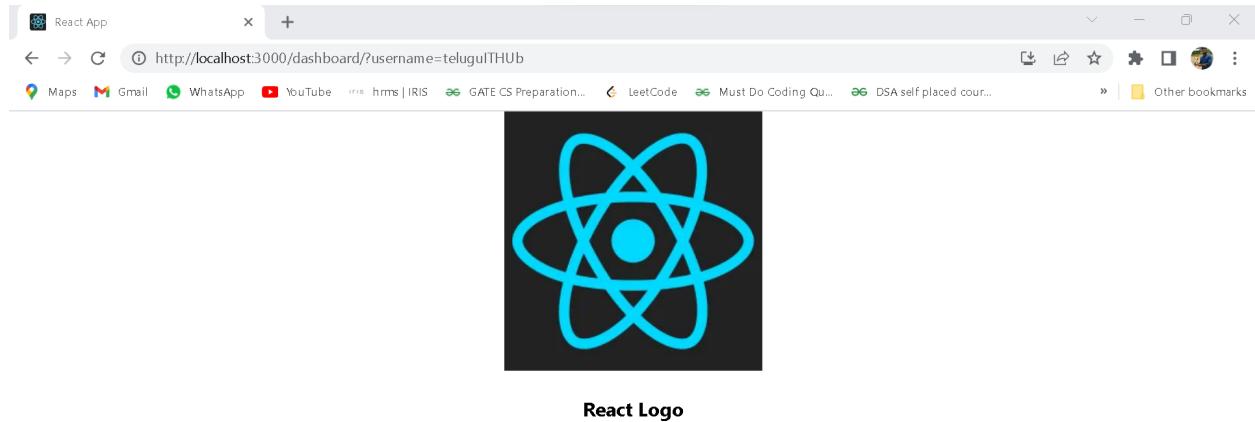
App.js

```
import React from 'react'
import ReactLogo from './react.png';

const App = () => {
  return (
    <div>
      <center>
        <img src={ReactLogo} height="250" width="auto" />
        <h3>React Logo</h3>
      </center>
    </div>
  )
}

export default App
```

O/P:



How to import audio in ReactJs

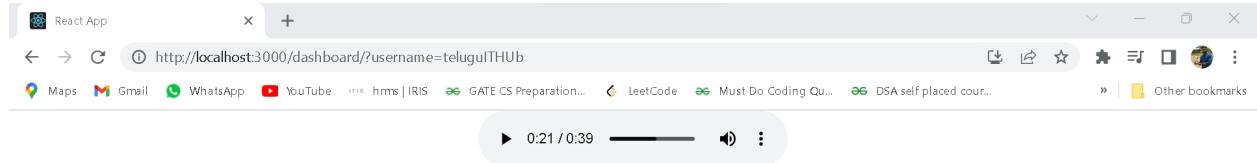
App.js

```
import React from 'react'
import Adipurush from './ram-siya-ram.mp3';

const App = () => {
  return (
    <div>
      <center>
        <audio controls>
          <source src={Adipurush} type='audio/ogg' />
        </audio>
      </center>
    </div>
  )
}

export default App
```

O/P:



How to import video in ReactJs

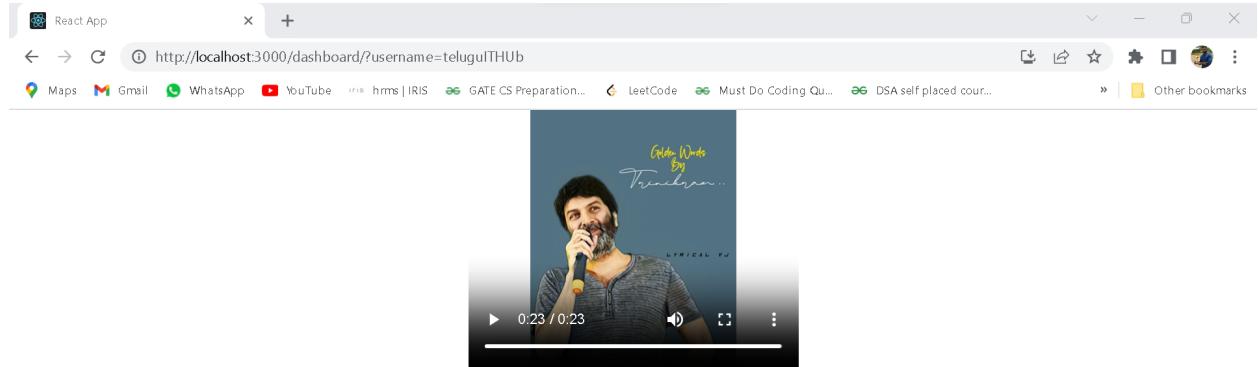
App.js

```
import React from 'react'
import VideoFile from './Trivikram.mp4';

const App = () => {
  return (
    <div>
      <center>
        <video width="320" height="250" controls>
          <source src={VideoFile} type='video/mp4' />
        </video>
      </center>
    </div>
  )
}

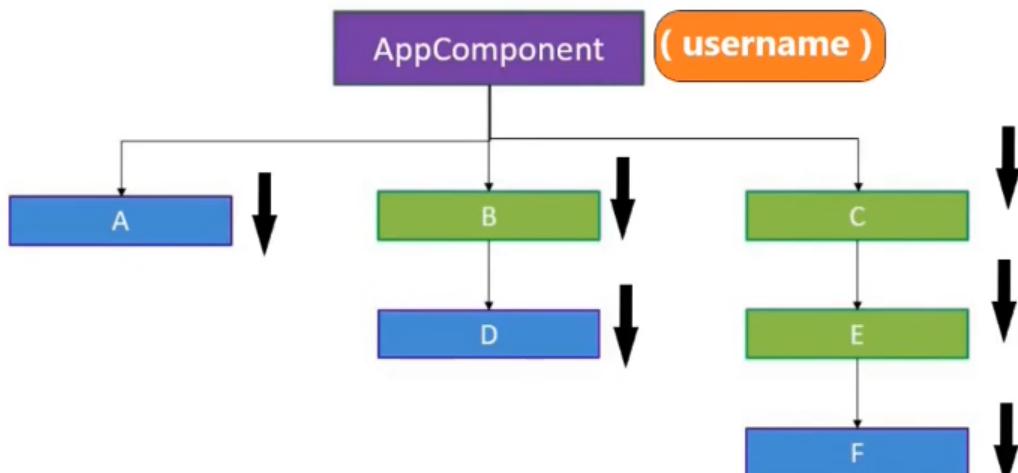
export default App
```

O/P:



useContext Hook in ReactJs

Context



Example1

Example2

Example3

→App.js to Component C sending Props, without intermediate B

App.js'

```
import React from 'react'
import ComponentC from './ComponentC'

export const UserContext = React.createContext();

const App = () => {
  return (
    <div>
      <center>
        <UserContext.Provider value={"Telugu Skillhub"}>
          <ComponentC></ComponentC>
        </UserContext.Provider>
      </center>
    </div>
  )
}

export default App
```

ComponentC.js

```
import React from 'react'
import { UserContext } from './App.js'

const ComponentC = () => {
  return (
    <div>
      <UserContext.Consumer>
        {value => <div> { value } </div>}
      </UserContext.Consumer>
    </div>
  )
}

export default ComponentC
```

```
        </UserContext.Consumer>
      </div>
    )
}

export default ComponentC
```

O/P:



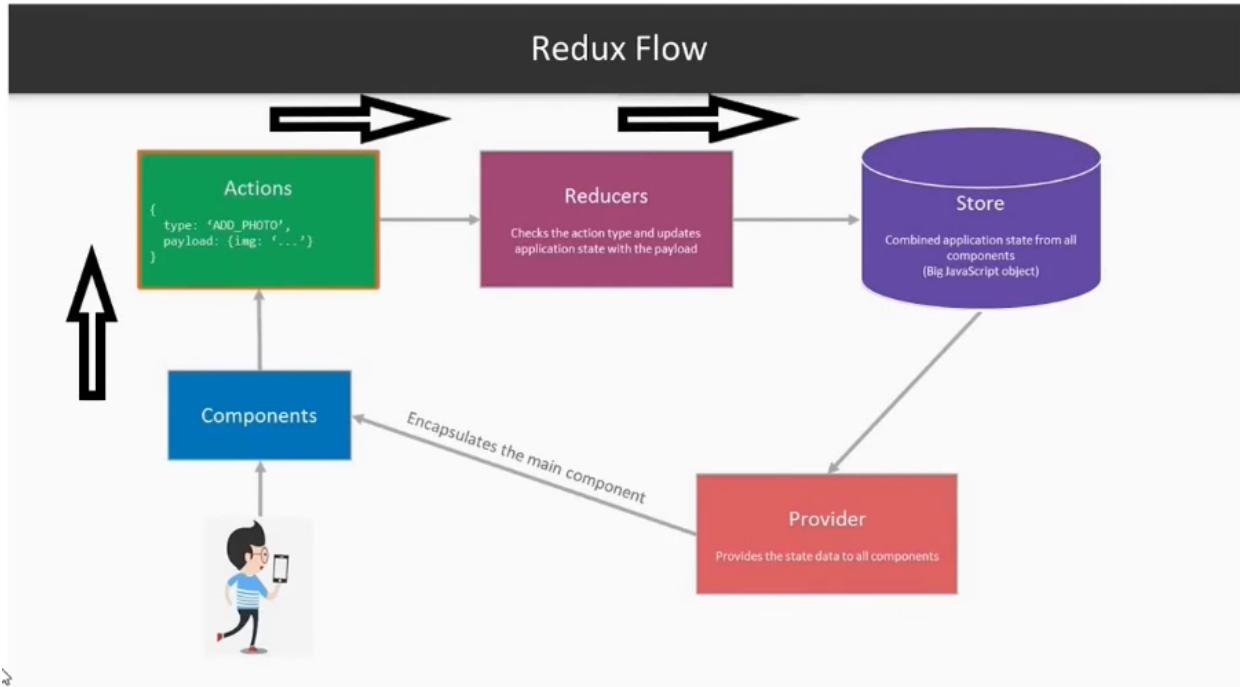
Redux in ReactJs

→ This complete video is available in 48 (#49)

React - Redux

Redux In React JS

It is all about **Managing state values in multiple components at a time, using redux store**



Dependencies To Be Installed For Redux

npm install redux

npm install react-redux

npm install redux-thunk

→

PS C:\Users\gudiw\Documents\tuts\sample> **npm install redux**

added 1 package, and audited 1495 packages in 2m

33 vulnerabilities (27 moderate, 6 high)

To address issues that do not require attention, run:

npm audit fix

To address all issues (including breaking changes), run:

npm audit fix --force

Run `npm audit` for details.

PS C:\Users\gudiw\Documents\tuts\sample> **npm install react-redux**

added 7 packages, and audited 1502 packages in 36s

235 packages are looking for funding

run `npm fund` for details

33 vulnerabilities (27 moderate, 6 high)

To address issues that do not require attention, run:

npm audit fix

To address all issues (including breaking changes), run:

npm audit fix --force

Run `npm audit` for details.

PS C:\Users\gudiw\Documents\tuts\sample> npm install redux-thunk

added 1 package, and audited 1503 packages in 7s

235 packages are looking for funding
run `npm fund` for details

33 vulnerabilities (27 moderate, 6 high)

To address issues that do not require attention, run:

npm audit fix

To address all issues (including breaking changes), run:

npm audit fix --force

Run `npm audit` for details.

App.js

```
import React from 'react'
import { Connect } from 'react-redux';
import Inc from './inc';

const App = ({count}) => {
  return (
    <div>
      <center>
        count from App Js Component : {count} <br></br>
        <Inc />
      </center>
    </div>
  )
}
```

```
export default App
```

inc.js

```
import React from 'react'
import { Connect } from 'react-redux'
import {increase, decrease} from './action';

const inc = ({count, increase, decrease}) => {
  return (
    <div>
      Count From Inc Component : {count} <br></br>
      <button onClick={() => increase()}>Increment</button>
      &nbsp;
      <button onClick={() => decrease()}>decrease</button>
    </div>
  )
}

export default inc
```

action.js

```
export const increase = () => dispatch =>{
  dispatch({
    type: "INCREMENT"
  })
}

export const decrease = () => dispatch =>{
  dispatch({
    type: "DECREMENT"
  })
}
```

```
}
```

reducer.js

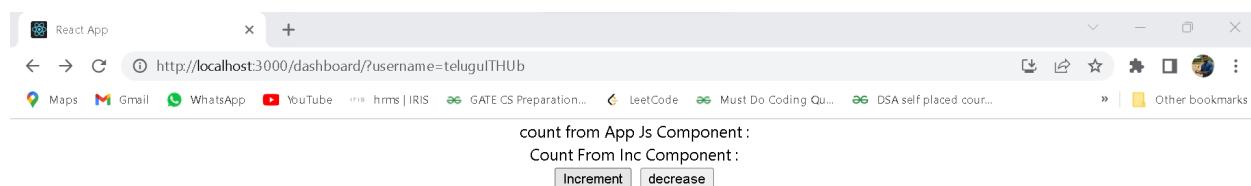
```
const initialState = 0;

function countReducer(state = initialState, action) {
  const {type, payload} = action;

  switch(type) {
    case "INCREMENT":
      return state+1;
    case "DECREMENT":
      return state-1;
    default:
      return state;
  }
}

export default countReducer;
```

O/P:



mapStateToProps in react redux

mapStateToProps In Redux

mapStateToProps In React-Redux

Function that you would use to provide the store data to your component.

App.js

```
import React from 'react'
import { Connect } from 'react-redux';
import Inc from './inc';

const App = ({count}) => {
  return (
    <div>
      <center>
        count from App Js Component : {count} <br></br>
        <Inc />
      </center>
    </div>
  )
}
```

```
//Destructure
const mapStateToProps = (state) => ({
  count : state
})
)

export default App
```

mapDispatchToProps in react redux

mapDispatchToProps In Redux

mapDispatchToProps In React-Redux

mapDispatchToProps is use to provide the **action creators as props** to your component.

inc.js

```
import React from 'react'
import { connect } from 'react-redux'
import {increase, decrease} from './action';

const inc = ({count, increase, decrease}) => {
```

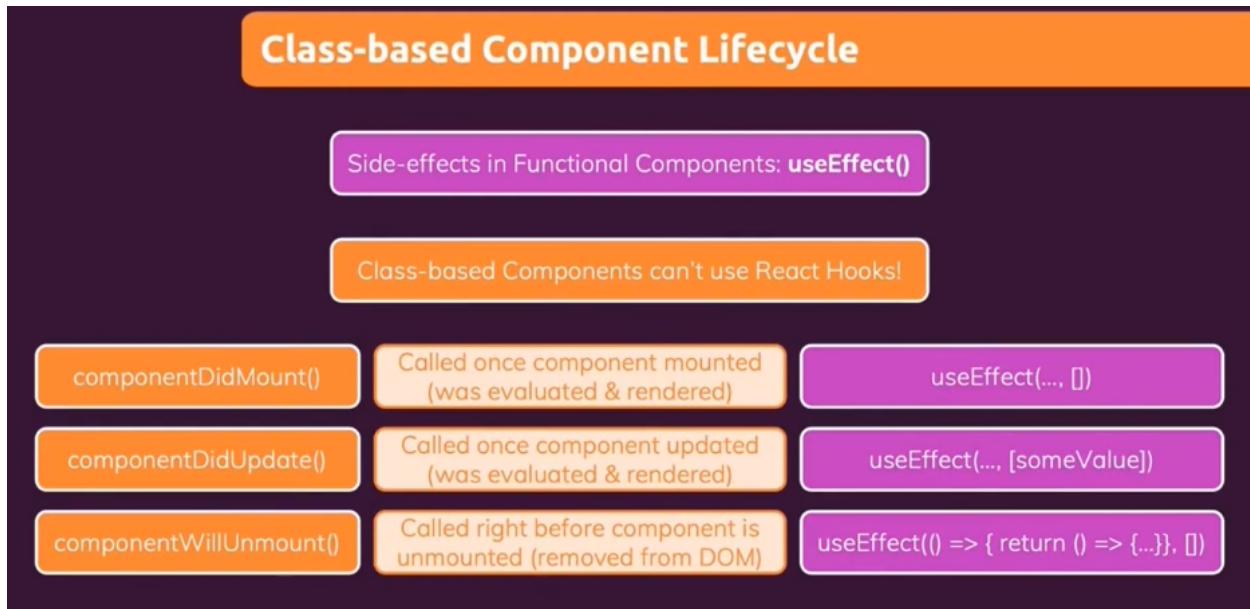
```
return (
  <div>
    Count From Inc Component : {count} <br></br>
    <button onClick={() => increase()}>Increment</button>
    &nbsp; &nbsp;
    <button onClick={() => decrease()}>decrease</button>
  </div>
)
}

const mapStateToProps = (state) =>({
  count : state
})

const mapDispatchToProps = (dispatch) =>{
  return{
    //dispatching plain actions
    increase: () => dispatch({ type: 'INCREMENT' }) ,
    decrease: () => dispatch({ type: 'DECREMENT' })
  }
}

export default connect(mapStateToProps,
mapDispatchToProps)(inc)
```

LifeCycle in Class-Based Components



App.js

```
import React from 'react';

class Container extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      show: true,
      count: 0
    };
  }
  delHeader = () =>{
    this.setState({show: false});
  }

  componentDidMount(){
    console.log('ComponentDidMount Method');
  }
}
```

```
}

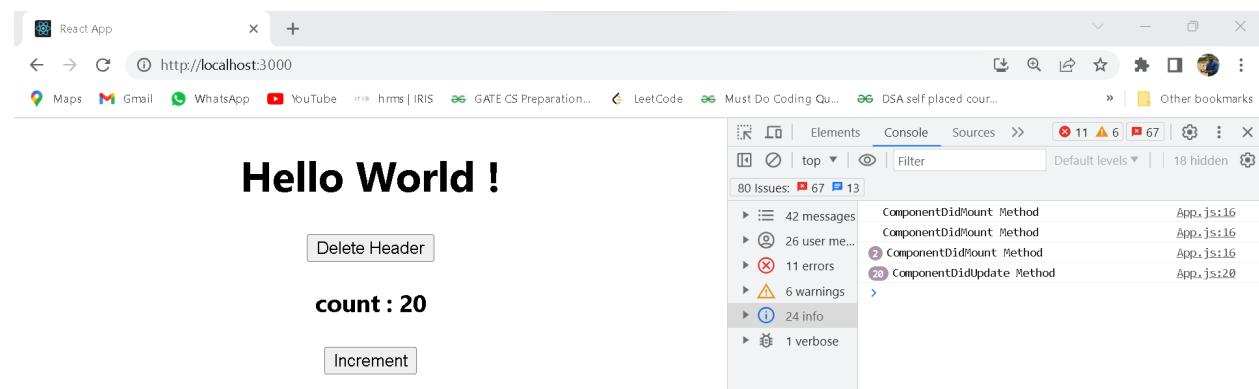
componentDidUpdate() {
  console.log('ComponentDidUpdate Method');
}

render() {
  let myheader;
  if(this.state.show) {
    myheader = <Child />;
  }
  return(
    <div>
      <center>
        {myheader}
        <button type='button'
onClick={this.delHeader}>Delete Header</button><br><br>
        <h3>count : {this.state.count}</h3>
        <button onClick={()=>this.setState({count :
this.state.count+1})}>Increment</button>
      </center>
    </div>
  );
}

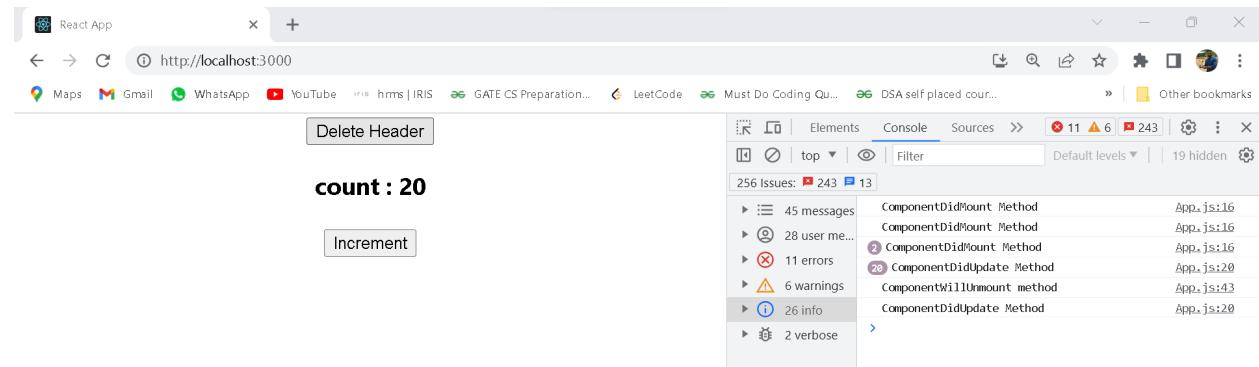
class Child extends React.Component {
  componentWillMount() {
    console.log('ComponentWillMount method');
  }
  render() {
    return(
      <h1>Hello World !</h1>
    )
  }
}
```

```
) ;  
}  
  
}  
  
export default Container;
```

O/P:



→After deletion of header



How To Use Bootstrap in ReactJs

<https://react-bootstrap.github.io/>

→Command to install

npm install react-bootstrap bootstrap

PS C:\Users\gudiw\Documents\tuts\sample> **npm install react-bootstrap bootstrap**

added 19 packages, and audited 1523 packages in 56s

237 packages are looking for funding
run `npm fund` for details

33 vulnerabilities (27 moderate, 6 high)

To address issues that do not require attention, run:

npm audit fix

To address all issues (including breaking changes), run:

npm audit fix --force

Run `npm audit` for details.

→The below command is added at Public folder **index.html**

```
<link  
rel="stylesheet"
```

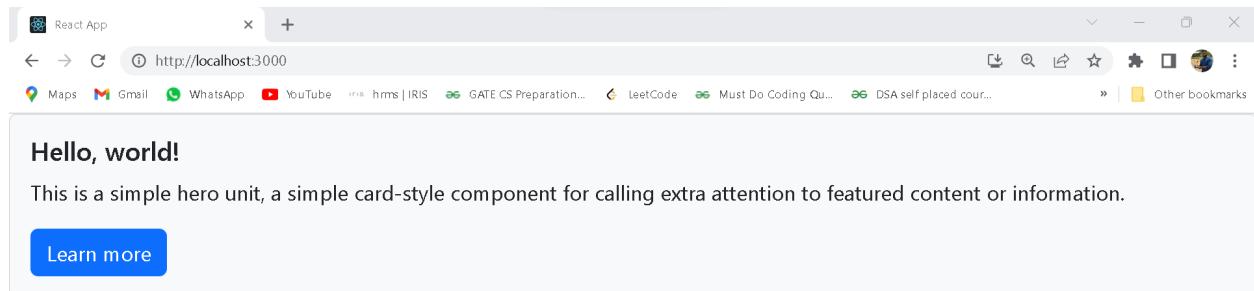
```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
```

```
integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLn  
uZ2cdeRhO02iuK6FUUVM"  
crossorigin="anonymous"  
/>>
```

App.js

```
import React from 'react';  
import { Card, Button } from 'react-bootstrap';  
  
const App = () => {  
  return (  
    <div>  
      <Card bg="light" text="dark">  
        <Card.Body>  
          <Card.Title>Hello, world!</Card.Title>  
          <Card.Text>  
            This is a simple hero unit, a simple card-style  
component for calling extra attention to featured content or  
information.  
          </Card.Text>  
          <Button variant="primary">Learn more</Button>  
        </Card.Body>  
      </Card>  
    </div>  
  );  
};  
  
export default App;
```

O/P:



How to Use Material UI in ReactJs

<https://mui.com/material-ui/getting-started/installation/>

→ Command to install

npm install @material-ui/core (old one)

npm install @mui/material @emotion/react @emotion/styled

PS C:\Users\gudiw\Documents\tuts\sample> **npm install @mui/material @emotion/react @emotion/styled**

added 30 packages, and audited 1554 packages in 1m

244 packages are looking for funding
run `npm fund` for details

33 vulnerabilities (27 moderate, 6 high)

To address issues that do not require attention, run:
npm audit fix

To address all issues (including breaking changes), run:
npm audit fix --force

Run `npm audit` for details.

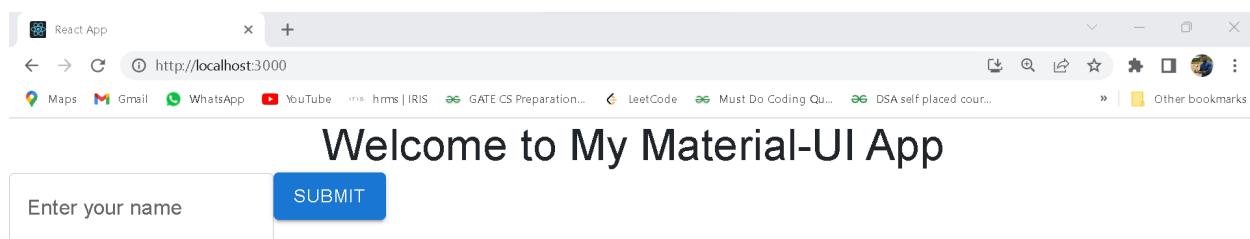
App.js

```
import React from 'react';
import { Button, TextField, Typography } from '@mui/material';

const App = () => {
  return (
    <div>
      <Typography variant="h4" component="h1" align="center">
        Welcome to My Material-UI App
      </Typography>
      <TextField label="Enter your name" variant="outlined" />
      <Button variant="contained" color="primary">
        Submit
      </Button>
    </div>
  );
}

export default App;
```

O/P:



Doubt clarification Session#1 in ReactJs

<http://localhost:3000/task>

<http://localhost:3000/task/reactjs>

App.js

```
import React from 'react';
import Navbar from './Navbar';
import "./App.css";
import {BrowserRouter, Routes, Route} from 'react-router-dom';
import Home from './Home';
import Dashboard from './Dashboard';
import About from './About';
import Task from './Task';
import IndividualTask from './IndividualTask';

const App = () => {
  return (
    <div>
      <BrowserRouter>
        <Navbar />
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/dashboard" element={<Dashboard />} />
          <Route path="/about" element={<About />} />
        </Routes>
      </BrowserRouter>
    </div>
  );
}

export default App;
```

```

        <Route path="/task" element={<Task />} />
        <Route path="/task/:id" element={<IndividualTask />} />
    />
    </Routes>
</BrowserRouter>
</div>
);
};

export default App;

```

Navbar.js

```

import React from 'react'
import { Link } from 'react-router-dom'

const Navbar = () => {
    return (
        <div className="navbar">
            <ul className="navbar-links">
                <Link to="/"><li>Home</li></Link>
                <Link to="/dashboard"><li>Dashboard</li></Link>
                <Link to="/about"><li>About</li></Link>
                <Link to="/task"><li>task</li></Link>
            </ul>
        </div>
    )
}

export default Navbar

```

Home.js

```
import React from 'react';
// import { useNavigate } from 'react-router-dom';

const Home = () => {
  // const navigate = useNavigate();

  return (
    <div>
      <center>
        <h4>Welcome To Home Page !!</h4>
        {/* <button onClick={() =>
navigate('/dashboard')}>Login</button> */}
      </center>
    </div>
  );
}

export default Home;
```

Dashboard.js

```
import React from 'react'

const Dashboard = () => {
  return (
    <div>
      <center>
        <h4>Welcome to Dashboard Page.</h4>
      </center>
    </div>
  );
}

export default Dashboard;
```

```
        </div>
    )
}

export default Dashboard
```

About.js

```
import React from 'react'

const About = () => {
    return (
        <div>
            <center>
                <h4>Your in About Page</h4>
            </center>
        </div>
    )
}

export default About
```

Task.js

```
import React, {useState, useEffect} from "react";
import {Link} from 'react-router-dom';

const Task = () => {
    const [data, setData] = useState([]);

    useEffect(()=>{
        fetch(`https://jsonplaceholder.typicode.com/todos`)
            .then(response => response.json())
            .then(data => setData(data))
    }, [])
}

export default Task
```

```

        .then(value =>setData(value))

    } , []
    return (
        <div>
            {data.length}>= 1 ? data.map(data =>{
                return <div style={{ "border": "1px solid black", "padding": "10px", "margin": "10px" }}><Link
to={`/task/${data.id}`}>{data.title}</Link></div>
            }
            ): null}
        </div>
    )
}

export default Task

```

IndividualTask.js

```

import React, { useState, useEffect } from 'react';
import { useParams } from 'react-router-dom';

const IndividualTask = () => {
    const { id } = useParams();
    const [individual, setIndividual] = useState(null);

    useEffect(() => {
        fetch(`https://jsonplaceholder.typicode.com/todos/${id}`)
            .then(response => response.json())
            .then(data => setIndividual(data));
    }, [id]);
}

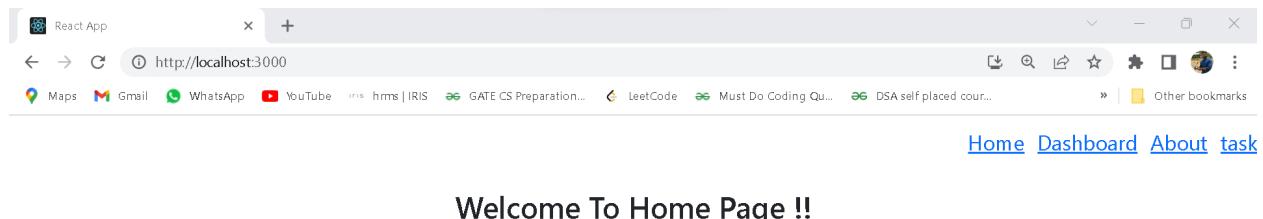
export default IndividualTask

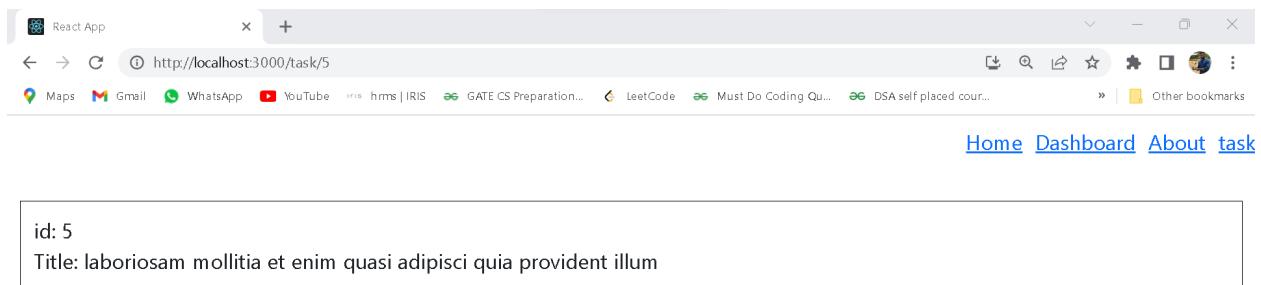
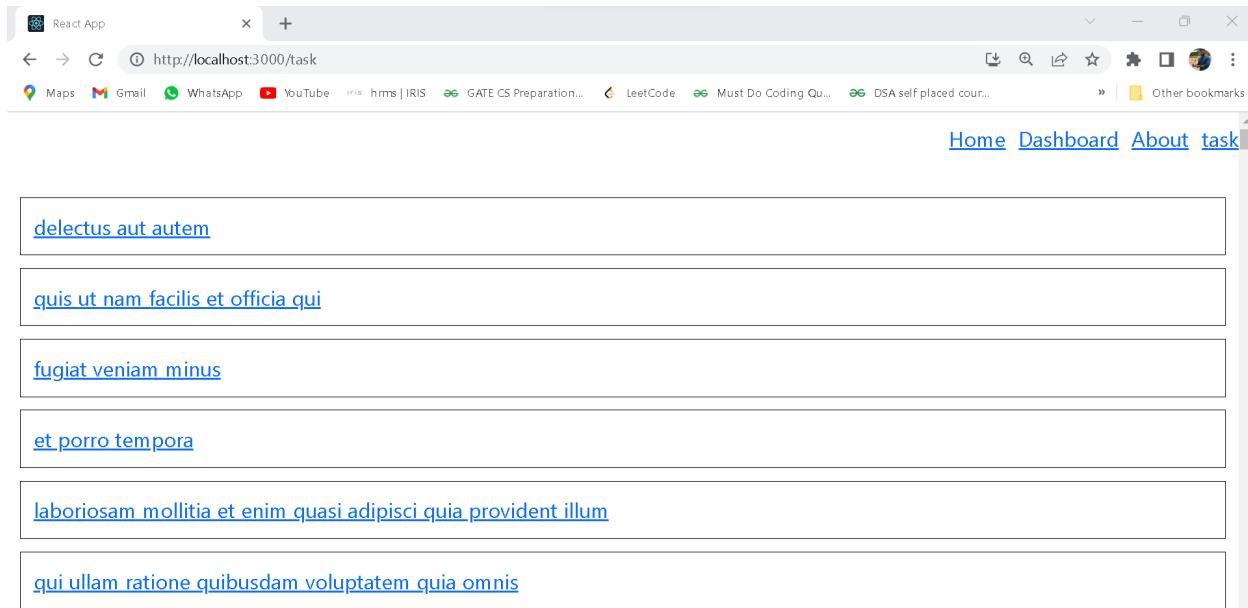
```

```
return (
  <div style={{ border: '1px solid black', padding: '10px',
margin: '10px' }}>
  {individual !== null ?
    <div>
      id: {individual.id} <br />
      Title: {individual.title} <br />
    </div>
    : null }
  </div>
);
};

export default IndividualTask;
```

O/P:





How to download files in ReactJs

App.js

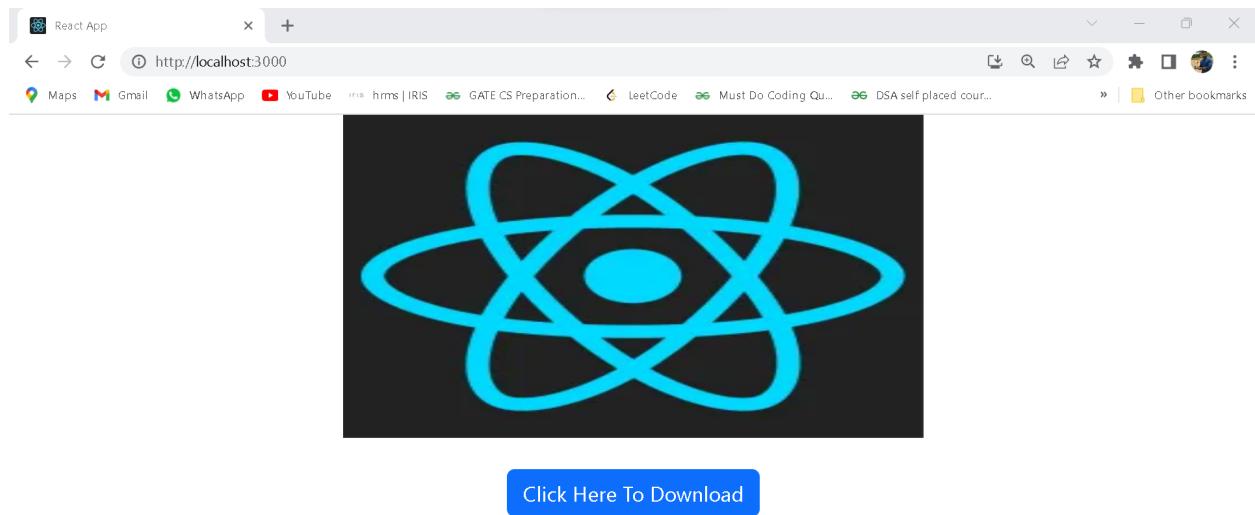
```
import React from 'react'
import Thumbnail from './react.png'

const App = () => {
  return (
    <div>
      <center>
```

```
        <img src={Thumbnail} alt='thumbnail' height="250"
width="450"></img><br /> <br />
        <a href={Thumbnail} className='btn btn-primary'
download='thumbnail'>Click Here To Download</a>
    </center>
</div>
)
}

export default App
```

O/P:



How to send emails in ReactJs using EmailJs



<https://www.emailjs.com/>

<https://www.emailjs.com/docs/>

→Examples →React

<https://www.emailjs.com/docs/examples/reactjs/>



https://www.emailjs.com/docs/

Maps Gmail WhatsApp YouTube hms | IRIS GATE CS Preparation... LeetCode Must Do Coding Qu... DSA self placed cour... Other bookmarks

EmailJS Home Docs Pricing Contact Us Sign In Sign Up Free

to spam?

Is it possible to send embedded attachments?

Can I send HTML emails?

What about plain text emails?

Can I send HTML from my code?

Can I add my domain to the allowlist?

Getting Started

Welcome to EmailJS documentation! Here are a few good places to start:

Take a look at [EmailJS Tutorial](#)

Get started with setting up your account via the [EmailJS dashboard](#)

Dive right in and [install EmailJS SDK](#) on your website

Browse through our [SDK](#) or [REST API](#)

Examples

React

Svelte

Vue.js

Angular

Wix Code

Copyright © 2023 EmailJS Pte Ltd. All rights reserved.

Follow us on [Twitter](#)

EmailJS uses cookies to enhance your experience. We use these for analytics purposes. By continuing to use our site, you agree to our [use of cookies](#) [Privacy Policy](#)



Your Email *

Password *



Forgot your password?

Sign In

<https://dashboard.emailjs.com/admin>

→ Click on “Add New Service”

The screenshot shows the EmailJS Admin Dashboard. At the top, there's a header bar with the EmailJS logo, a quota indicator ("QUOTA REMAINING: 200 EMAILS"), and user information ("Welcome, Raghava"). Below the header, a sidebar on the left lists "Email Services", "Email Templates", "Contacts", and "Email History". The main content area is titled "Email Services" and features a prominent blue button labeled "+ Add New Service". A message at the bottom of this area says, "How will we send emails? Press Add New Service to add your first service!". The browser's address bar shows the URL "https://dashboard.emailjs.com/admin".

→ Selecting “Gmail”

Select Service X

Personal Services ?

 Gmail	 AOL	 Fastmail	 iCloud	 Mail.ru
 Outlook	 Outlook v2	 Yahoo	 Yandex	 Zoho
 SMTP server				

Transactional Services ?

→Select “Connect Account”

Config Service X

 **Gmail**
Personal Service | 500 emails per day ⓘ

Name *

Service ID *
 

Connect Account

 Allow "Send email on your behalf" permission during connection.
Both Gmail and Google Apps accounts are supported.

Send test email to verify configuration

>EmailJS QUOTA REMAINING: 200 EMAILS Welcome, Raghava Docs Support Sign Out

 Email Services  Add New Service

 Email Templates

 Contacts

 Email History

Email Services

 **Gmail**
Service ID: `service_gf3vrpc` DEFAULT   

The screenshot shows the EmailJS dashboard at <https://dashboard.emailjs.com/admin/templates/2mw04iq>. The left sidebar has 'Email Templates' selected. The main area is titled 'My Default Template' with a quota of 200 emails remaining. The template content includes a subject line with {{Subject}} and a message body with placeholders like {{to_name}}, {{from_name}}, and {{message}}. On the right, there are fields for 'To Email' (gudiwadaraghava999@gmail.com), 'From Name' (ReactJS Testing), and 'From Email' (checkbox for 'Use Default Email Address'). Buttons for 'Playground', 'Test It', and 'Save' are at the top right.

The screenshot shows the EmailJS dashboard at <https://dashboard.emailjs.com/admin/templates>. The left sidebar has 'Email Templates' selected. The main area is titled 'Email Templates' with a quota of 200 emails remaining. It shows a list of templates, with 'My Default Template' (Template ID: template_90snd3p) highlighted. A 'Create New Template' button is visible above the list. Action buttons for edit, delete, and trash are shown to the right of each template entry.

→Command to install EmailJS

npm install emailjs-com

PS C:\Users\gudiw\Documents\tuts\sample> **npm install emailjs-com**
npm WARN deprecated emailjs-com@3.2.0: The SDK name changed to
@emailjs/browser

added 1 package, and audited 1555 packages in 27s

244 packages are looking for funding
run `npm fund` for details

7 vulnerabilities (1 moderate, 6 high)

To address issues that do not require attention, run:

To address all issues (including breaking changes), run:
npm audit fix --force

Run `npm audit` for details.

App.js

```
import React, {useState} from 'react'
import emailjs from 'emailjs-com';

const App = () => {
  const [data, setData] = useState({
    subject:'',
    email:'',
    message:''
  })

  const {subject, email, message} = data;

  const changeHandler = e =>{
    setData({...data, [e.target.name]:e.target.value})
  }

  const submitHandler = e =>{
    e.preventDefault();
    emailjs.send('service_id', 'template_id', data, 'user_email')
    .then((result) => console.log(result))
    .catch((error) => console.error(error))
  }
}

export default App;
```

```
emailjs.sendForm('service_gf3vrpc', 'template_90snd3p',
e.target, 'QLzVxzVR1Uj10bJ8K')
    .then((result) => {
        console.log(result.text);
    }, (error) => {
        console.log(error.text);
    });
    e.target.reset();
};

return (
    <div style={{ display: 'flex', justifyContent: 'center',
alignItems: 'center', height: '100vh' }}>
    <form onSubmit={submitHandler} autoComplete='off'>
        <label>Subject:</label><br />
        <input type='text' name="subject" value={subject}
onChange={changeHandler} placeholder='Subject'></input><br />
        <label> Email:</label><br />
        <input type='email' name='email' value={email}
onChange={changeHandler} placeholder='Email'></input><br />
        <label>Message:</label><br />
        <input type='text' name='message' value={message}
onChange={changeHandler} placeholder='Message'></input><br
/><br />
        <input type='submit' name="submit"></input>
    </form>
</div>
)
}

export default App
```

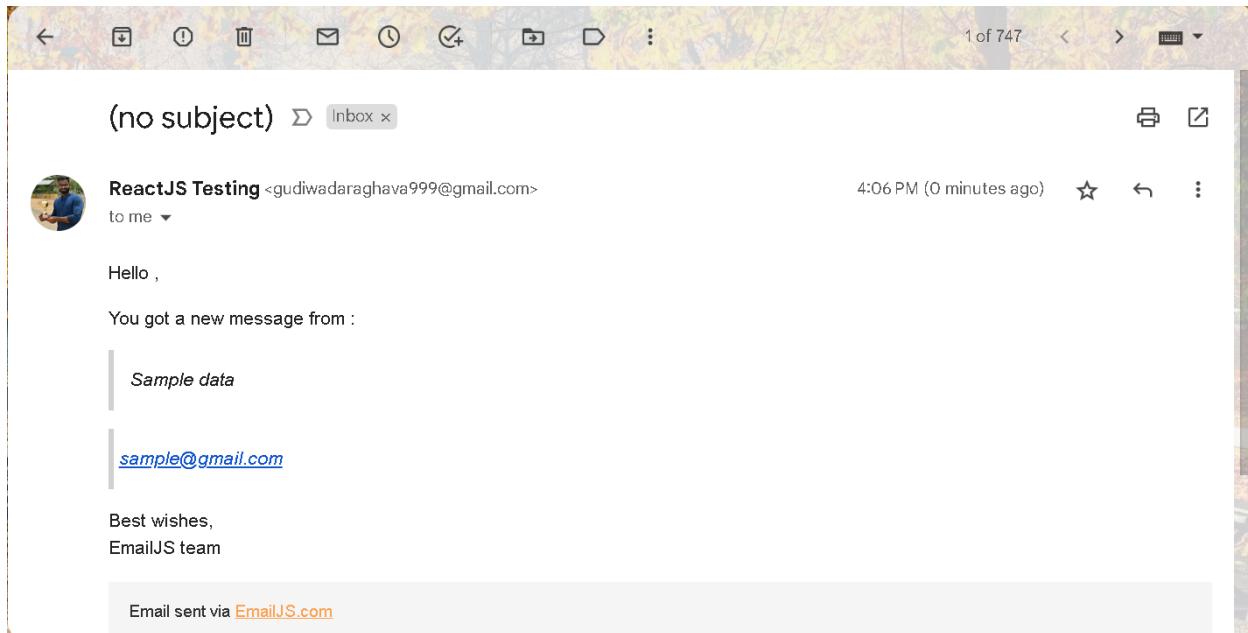
O/P:



Subject:

Email:

Message:



Search Filter in ReactJs

App.js

```
import React, {useState} from 'react'
import Data from './city.json'

const App = () => {
  const [search, setSearch] = useState('');

  return (
    <div>
      <center>
        <h4>Enter your City: </h4>
        <input type='text' value={search} onChange={(e) =>
setSearch(e.target.value)} /><br />
        {Data.filter(city =>
          city.name.toLowerCase().includes(search.toLowerCase())) .map(city =>{
```

```
        return <div style={{ "border": "1px solid black",  
"padding": "10px", "margin": "10px", "maxWidth": "70%" }}>  
            {city.name}  
        </div>  
    } )}  
    </center>  
</div>  
)  
}  
  
export default App
```

city.json

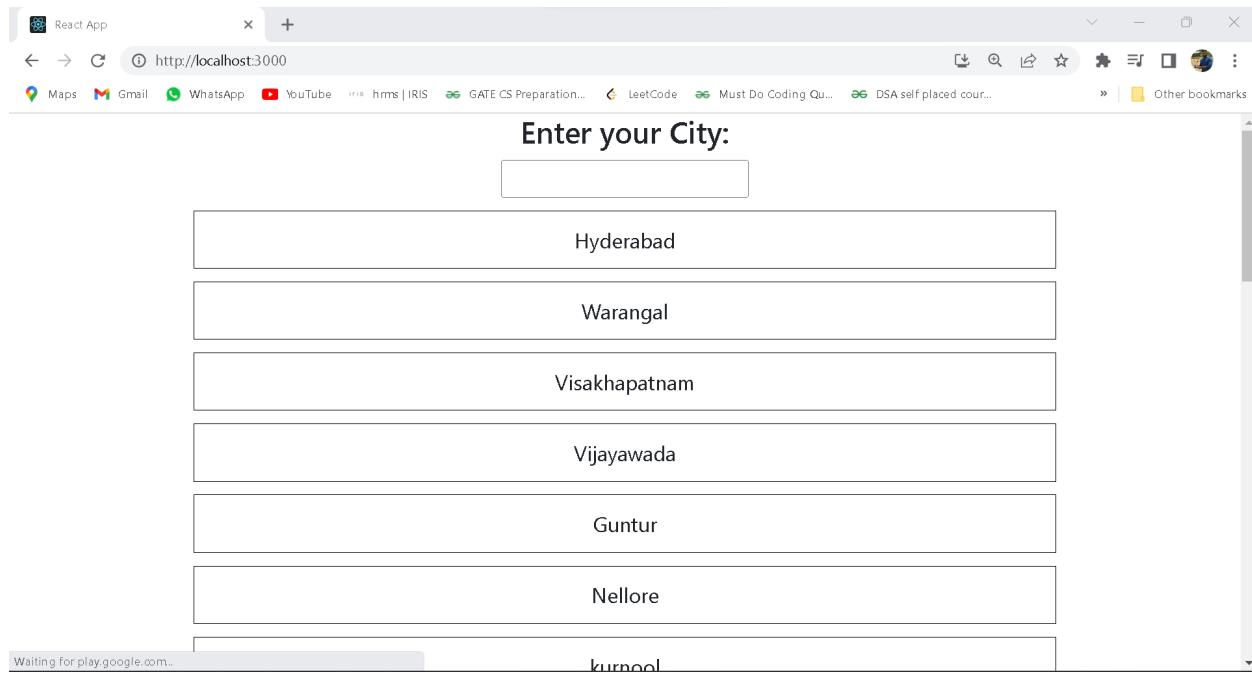
```
[  
 {  
     "id" : "1",  
     "name" : "Hyderabad"  
 },  
 {  
     "id" : "2",  
     "name" : "Warangal"  
 },  
 {  
     "id" : "3",  
     "name" : "Visakhapatnam"  
 },  
 {  
     "id" : "4",  
     "name" : "Vijayawada"  
 },  
 {  
     "id" : "5",  
     "name" : "Kurnool"  
 }]
```

```
        "name" : "Guntur"
    } ,
{
    "id" : "6",
    "name" : "Nellore"
},
{
    "id" : "7",
    "name" : "kurnool"
},
{
    "id" : "8",
    "name" : "kadapa"
},
{
    "id" : "9",
    "name" : "Tirupati"
},
{
    "id" : "10",
    "name" : "Anantapuram"
},
{
    "id" : "11",
    "name" : "Ongole"
},
{
    "id" : "12",
    "name" : "Chittoor"
},
{
    "id" : "13",
    "name" : "Machilipatnam"
```

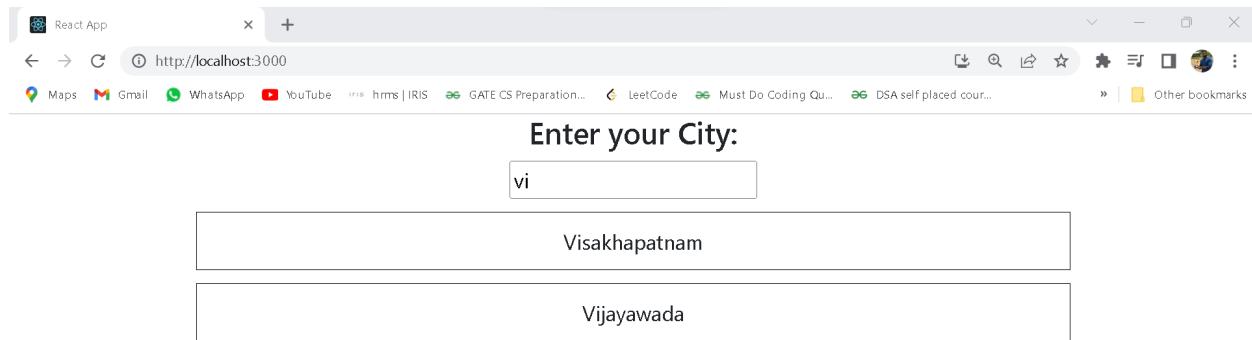
```
},
{
    "id" : "14",
    "name" : "Chilakaluripeta"
},
{
    "id" : "15",
    "name" : "Srikakulam"
},
{
    "id" : "16",
    "name" : "Bhimavaram"
},
{
    "id" : "17",
    "name" : "Hindupur"
},
{
    "id" : "18",
    "name" : "Khammam"
},
{
    "id" : "19",
    "name" : "Peddapalli"
},
{
    "id" : "20",
    "name" : "Suryapet"
},
{
    "id" : "21",
    "name" : "Karimnagar"
},
```

```
[  
    {  
        "id" : "22",  
        "name" : "Adilabad"  
    },  
    {  
        "id" : "23",  
        "name" : "Nagrkurnool"  
    },  
    {  
        "id" : "24",  
        "name" : "Nalgonda"  
    },  
    {  
        "id" : "25",  
        "name" : "Rajamma Sircilla"  
    },  
    {  
        "id" : "26",  
        "name" : "Sangareddy"  
    }  
]
```

O/P:



→Search



React Interview Questions and Answers

1. What is **React.js**?

React is front-end JavaScript library and used for building component based user interfaces UI.

It is maintained by Facebook.

2. What is **JSX** in React.js?

**JSX stands for JavaScript XML.
JSX allows us to write HTML in React.**

ex : const tag = <h1> text </h1>

3. What is a **useContext hook**?

useContext hook allows passing data to children elements without using redux.

4. What is **redux** in React?

Redux is an open-source JavaScript library and used to manage state values globally.

5. What is **Redux Thunk** in React?

Redux Thunk **middleware** allows you to write action creators that **return a function** instead of an **action**.

6. Difference between **useContext** and **redux**? (imp)

useContext

useContext is a hook.

It is used to share data.

It re-renders all components whenever there is any update in the provider's value prop.

Recommended for small applications.

Redux

Redux is a state management library.

It used to manage data & state.

It only re-render the updated components.

Perfect for larger applications.

7. What is the use of react hooks?

Hooks are functions used to manage both **React state** and **lifecycle** features from function components.

Hooks don't work inside classes !

8. What is **Mounting** and **Unmounting** in React?

Mounting is the process of outputting or adding a component to the DOM

Unmounting is the process of removing or blocking them from the DOM

9. Which **DOM** is Faster in React?

Virtual DOM is extremely **faster** because it is an abstraction of the HTML DOM and lightweight.

Real DOM is bit **slower** because of update the content, structure, and style of a document

48. Complete React Redux (#49)

Redux is an **open-source** **JavaScript library** for **managing** **application state**.

Packages required to use redux

npm install redux

npm install react-redux

npm install redux-thunk

npm install redux-devtools-extension

→First three is already installed

```
PS C:\Users\gudiw\Documents\tuts\sample> npm install  
redux-devtools-extension
```

```
npm WARN deprecated redux-devtools-extension@2.13.9: Package moved to  
@redux-devtools/extension.
```

added 1 package, and audited 1504 packages in 23s

```
235 packages are looking for funding  
run `npm fund` for details
```

33 vulnerabilities (27 moderate, 6 high)

To address issues that do not require attention, run:

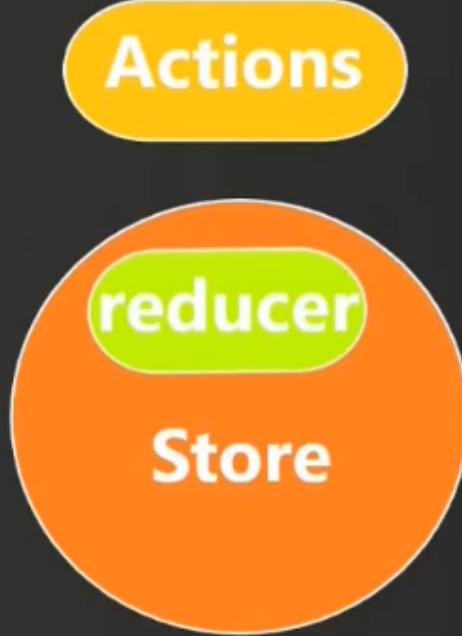
```
npm audit fix
```

To address all issues (including breaking changes), run:

```
npm audit fix --force
```

Run `npm audit` for details.

Major Terms In Redux

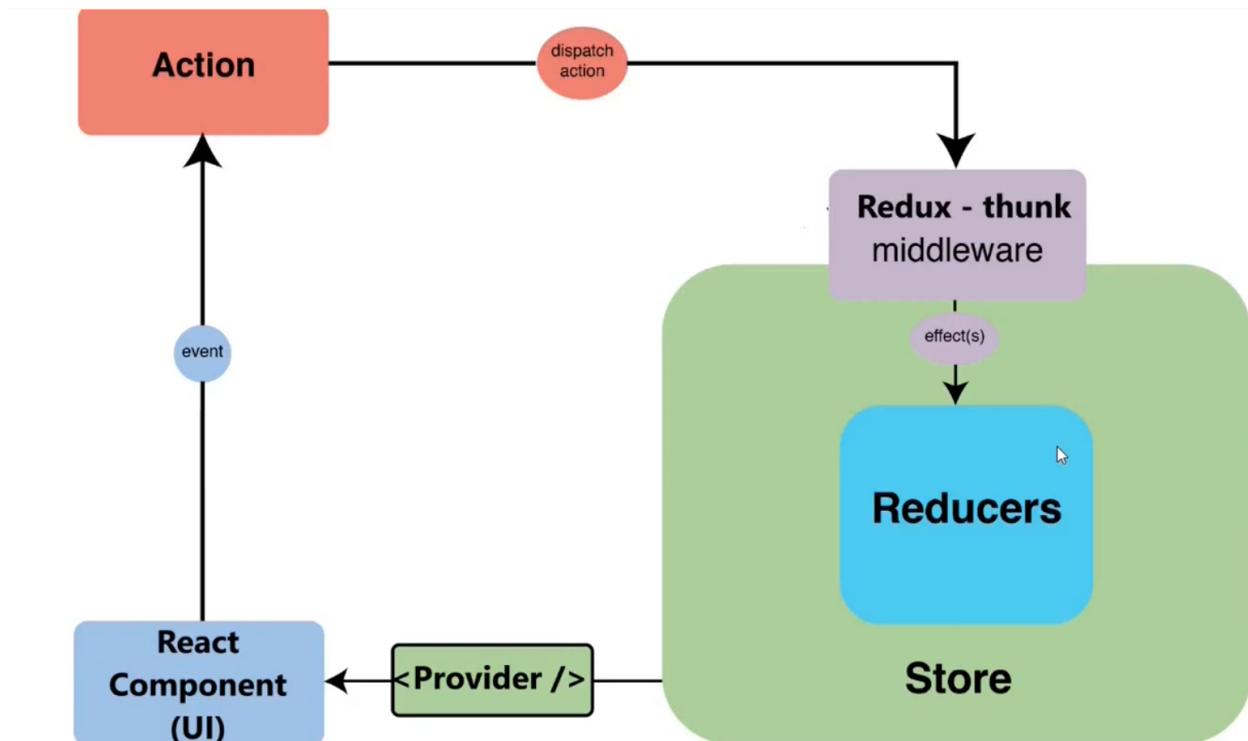
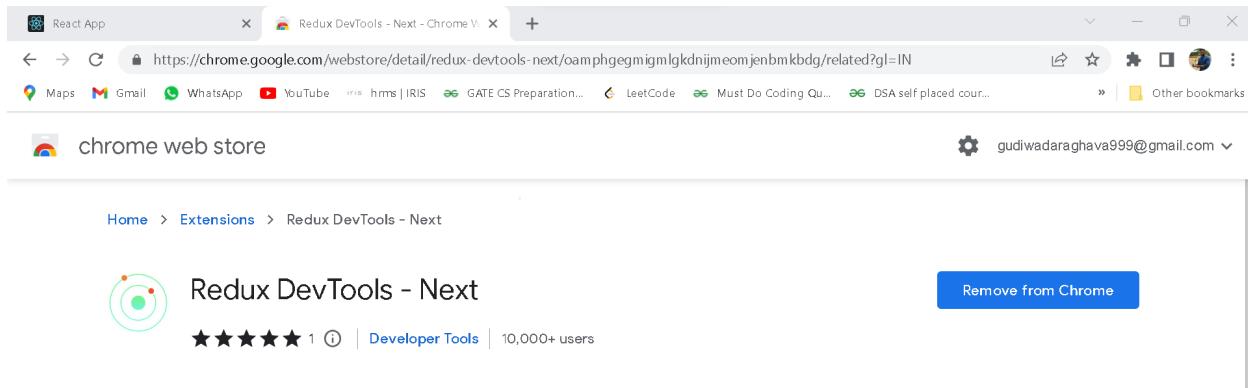


#teluguskillhub

- Store is used to store the state values
- reducer used to manage the state value.
- request will be coming from action

“Provider” is a module in the “react-redux”.

→ chrome webstore add extension “**Redux DevTools**”



index.js

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

```

import reportWebVitals from './reportWebVitals';
import store from './store';
import { Provider } from 'react-redux';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <Provider store={store}>
      <App />
    </Provider>
  </React.StrictMode>
);

reportWebVitals();

```

App.js

```

import React from 'react'
import { connect } from 'react-redux';
import { IncAction } from './actions';
import { DecAction } from './actions';

const App = ({local_variable, IncAction, DecAction}) => {
  return (
    <div>
      <center>
        <h1>count from App Js Component : {local_variable}</h1> <br></br>
        <button onClick={IncAction} >Increment</button>
        <button onClick={DecAction} >Decrement</button>
      </center>
    </div>
  );
}

export default connect(state => ({local_variable: state.local_variable}), {IncAction, DecAction})(App);

```

```

        </div>
    )
}

//Destructure
const mapStateToProps = (state) =>({
    local_variable : state
})

export default connect(mapStateToProps, {IncAction, DecAction})(App);

```

actions.js

```

import React from 'react'
import { connect } from 'react-redux';
import { IncAction } from './actions';
import { DecAction } from './actions';

const App = ({local_variable, IncAction, DecAction}) => {
    return (
        <div>
            <center>
                <h1>count from App Js Component :
{local_variable}</h1> <br><br>
                <button onClick={IncAction} >Increment</button>
                <button onClick={DecAction} >Decrement</button>
            </center>
        </div>
    )
}

//Destructure
const mapStateToProps = (state) =>({
    local_variable : state
})

```

```
)}

export default connect(mapStateToProps, {IncAction,
DecAction})(App);
```

reducer.js

```
import {createStore, applyMiddleware} from 'redux';
import { composeWithDevTools } from
'redux-devtools-extension';
import thunk from 'redux-thunk';
import reducer from './reducer';

const middleware = [thunk]

const store = createStore(
  reducer,
  composeWithDevTools(applyMiddleware(...middleware))
);

export default store;
```

store.js

```
import {createStore, applyMiddleware} from 'redux';
import { composeWithDevTools } from
'redux-devtools-extension';
import thunk from 'redux-thunk';
import reducer from './reducer';

const middleware = [thunk]
```

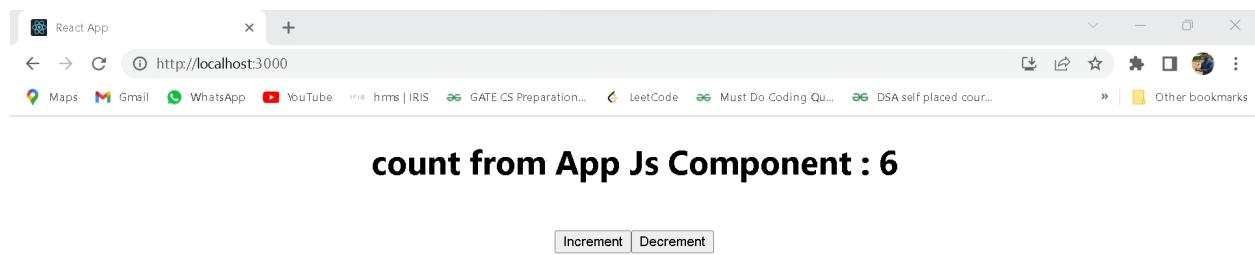
```

const store = createStore(
  reducer,
  composeWithDevTools(applyMiddleware(...middleware))
);

export default store;

```

O/P:



→Using Payload →Increment (3) & Decrement (2)

App.js

```

import React from 'react'
import { connect } from 'react-redux';
import { IncAction } from './actions';
import { DecAction } from './actions';

const App = ({local_variable, IncAction, DecAction}) => {
  return (
    <div>
      <center>
        <h1>count from App Js Component :
        {local_variable}</h1> <br><br>

```

```

        <button onClick={() => IncAction(3)}>
>Increment</button>
        <button onClick={() => DecAction(2)}>
>Decrement</button>
    </center>
</div>
)
}
//Destructure
const mapStateToProps = (state) =>({
    local_variable : state
})

export default connect(mapStateToProps, {IncAction,
DecAction})(App);

```

actions.js

```

export const IncAction = (value) => async dispatch =>{
    dispatch({
        type:"INCREMENT",
        payload: value
    })
}

export const DecAction = (value) => async dispatch =>{
    dispatch({
        type:"DECREMENT",
        payload: value
    })
}

```

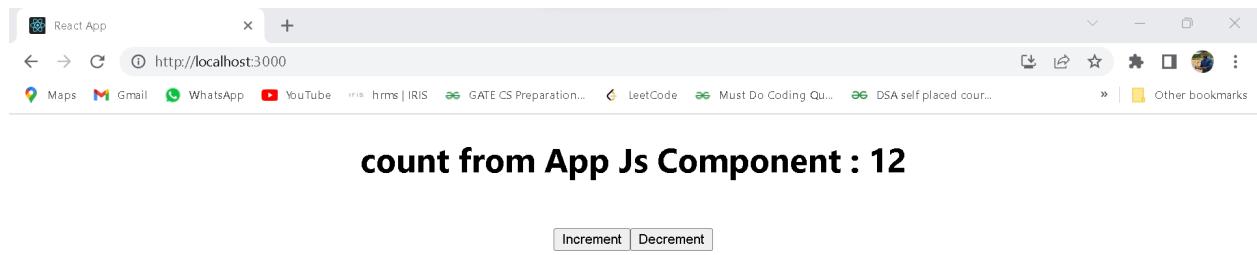
reducer.js

```
const count = 0;

export default function reducer(state = count, action) {
  const {type, payload} = action;

  switch(type) {
    case "INCREMENT":
      return state+payload;
    case "DECREMENT":
      return state-payload;
    default:
      return state;
  }
}
```

O/P:



Class Based Component In React

16.8 version → React hooks (State Management & Life Cycle) → Functional Level Components Used

“rcc” →Using the extension

→Declaring Class Based Component

```
import React, { Component } from 'react'

export default class App extends Component {
  render() {
    return (
      <div>

        </div>
      )
    }
}
```

<https://jsonplaceholder.typicode.com/todos> →Fake API's

App.js

```
import React, { Component } from 'react'
import Display from './Display'

export default class App extends Component {
  state = {
    name: "Skillhub",
    products: []
  }
  componentDidMount() {
    fetch('https://jsonplaceholder.typicode.com/todos')
      .then(res => res.json())
      .then(data => this.setState({products: data}))
      .catch(err => console.log(err))
  }
}
```

```

        }

      render() {
        return (
          <div>
            <Display data={this.state.products} />
            {/* {this.state.products.map((product) => <li
key={product.id}>{product.title}</li>) } Printing in this page
*/}
            {/* <Display data = {this.state.name} /> Props */}
            <br />
            <button onClick={() => this.setState({name:"Telugu
Skillhub"})}>Change Name</button>

          </div>
        )
      }
    }
  
```

Display.js

```

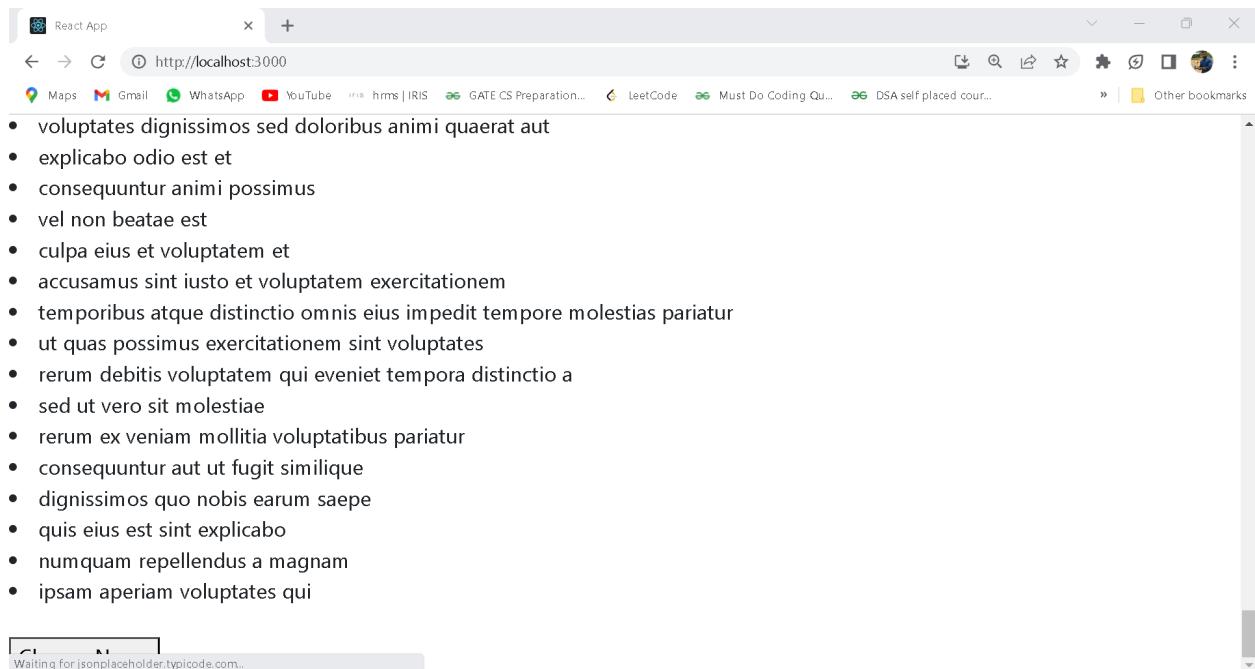
import React, { Component } from 'react'

export default class Display extends Component {
  render() {
    return (
      <div>
        {/* <h1>{this.props.data}</h1> */}
        {this.props.data.map((product) => (
          <li key={product.id}>{product.title}</li>
        )))
      </div>
    )
  }
}

```

```
}
```

O/P:



What is useRef Hook in React

useState will re-render when the content change and update in UI .

useRef doesn't notify you when its content changes. Mutating the (.current) property doesn't cause a re-render .

→onChange Event handler are used when it is working with {useState}

→when using with {useRef} no onChange handler is needed

App.js

```
import React, {useRef, useEffect} from 'react'

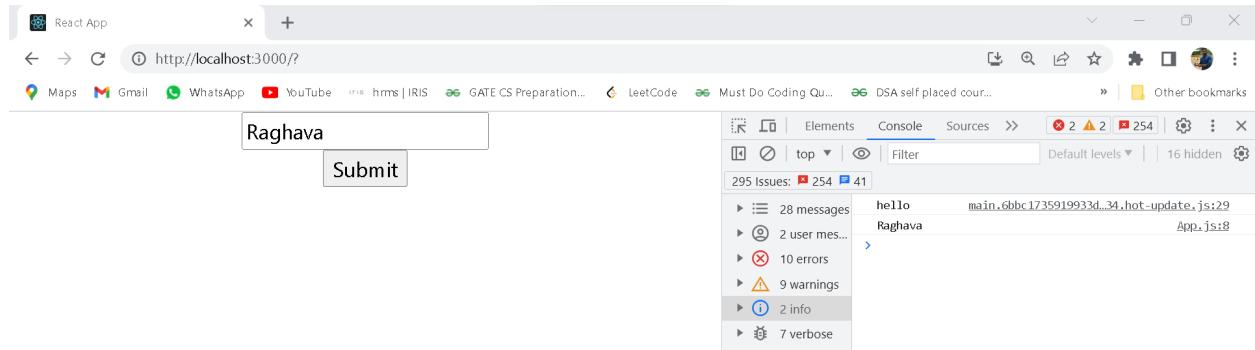
const App = () => {
  const data = useRef(null);

  const submitHandler = e =>{
    e.preventDefault();
    console.log(data.current.value);
  }

  //Auto Focusing -->Cursor placing at where ever to type
  useEffect(()=>{
    data.current.focus();
  }, [])
  return (
    <div>
      <center>
        <form onSubmit={submitHandler}>
          <input ref = {data} type='text' placeholder='Enter
your name' /> <br />
          <input type='submit' />
        </form>
      </center>
    </div>
  )
}

export default App
```

O/P:



What is useReducer Hook in React

→ It has workflow as Redux()

Redux -> use to store state values in redux-store

useReducer -> use to store state values to local const variable

```
const [state, dispatch] =  
  useReducer(reducer, initialState);
```

App.js

```
import React, {useReducer} from "react";

const initialState = {count: 0};

function reducer(state, action) {
  switch (action.type) {
    case 'increment':
      return {count: state.count + 1};
    case 'decrement':
      return {count: state.count - 1};
    default:
      throw new Error();
  }
}

function App() {
  const [state, dispatch] = useReducer(reducer, initialState);

  return (
    <center>
      count: {state.count}
      <button onClick={() => dispatch({type: 'decrement'})}>-</button>
      <button onClick={() => dispatch({type: 'increment'})}>+</button>
    </center>
  )
}

export default App;
```

O/P:

→**Initial**



→**'increment'**



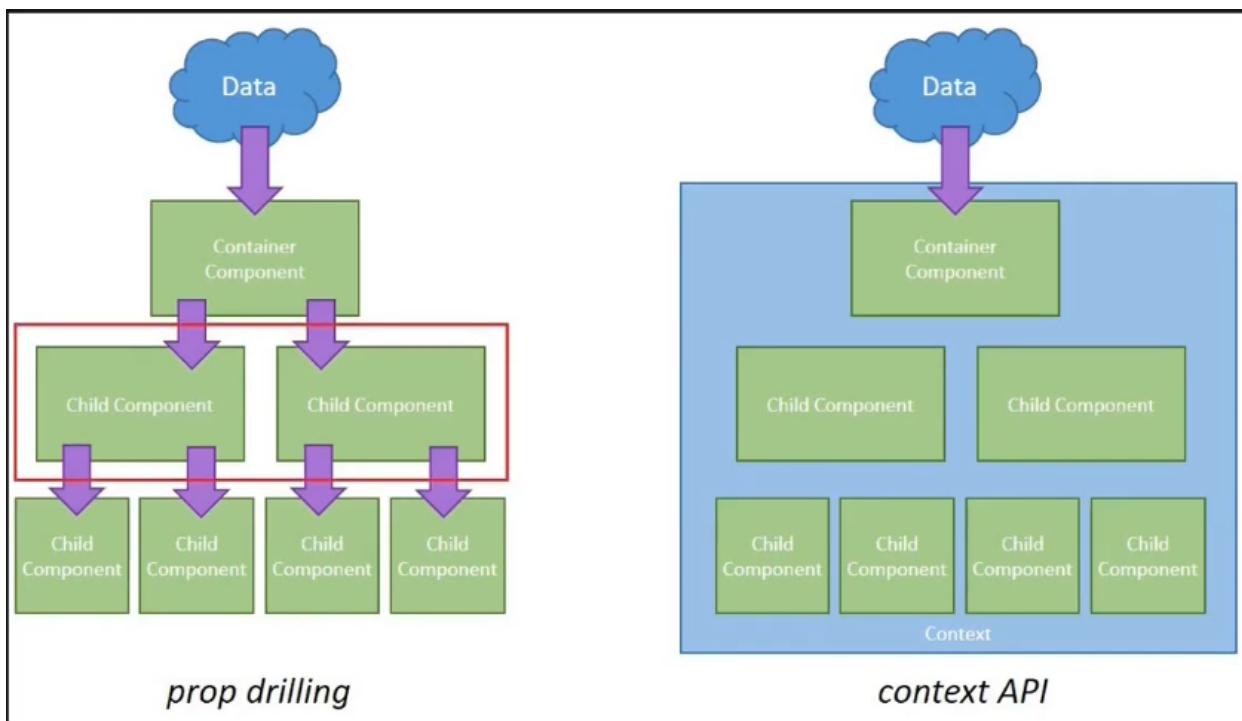
→**'decrement'**

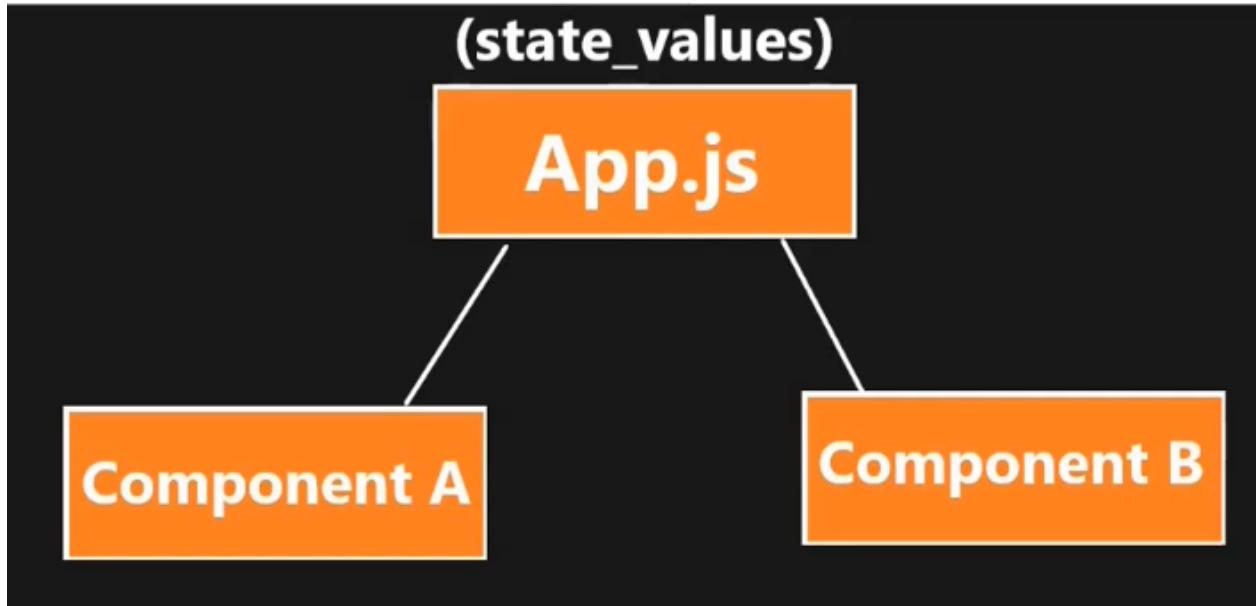


Context API in ReactJs

Context API is a kind of feature used to share data with multiple components.

Without passing the data through **props** manually.





App.js

```
import React, {createContext, useState} from 'react'
import ComponentA from './ComponentA'
import ComponentB from './ComponentB'

export const store = createContext();

const App = () => {
  const [data, setData] = useState(0);

  return (
    <store.Provider value={[data, setData]}>
      <center>
        <ComponentA />
        <ComponentB />
        <button onClick={()=>
          setData(data+1)}>Increment</button>
      </center>
    </store.Provider>
  );
}
```

```
)  
}  
  
export default App
```

ComponentA.js

```
import React, {useContext} from 'react'  
import {store} from './App';  
  
const ComponentA = () => {  
  const [data, setData] = useContext(store);  
  return (  
    <div className='card' style={{'maxWidth': "60%",  
"margin": "10px"} }>  
      <div className='card-body'>  
        Component A {data}  
      </div>  
    </div>  
  )  
}  
  
export default ComponentA
```

ComponentB.js

```
import React, {useContext} from 'react'  
import {store} from './App';  
  
const ComponentB = () => {  
  const [data, setData] = useContext(store);  
  return (
```

```

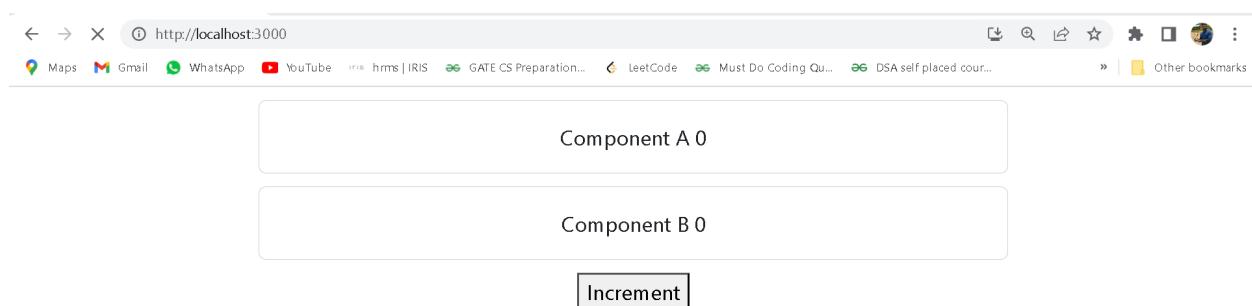
    <div className='card' style={{ "maxWidth": "60%", "margin": "10px" }}>
      <div className='card-body'>
        Component B {data}
      </div>
    </div>
  )
}

export default ComponentB

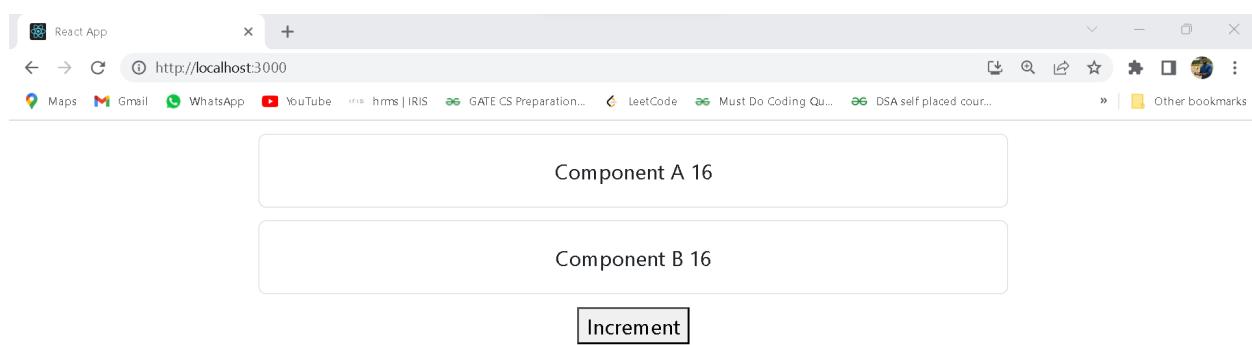
```

O/P:

→Initially



→After Increment



App.js

```
import React, {createContext, useState} from 'react'
import Count from './Count';
import Display from './Display';

export const store = createContext();

const App = () => {
  const [data, setData] = useState([
    {
      brandname : "Nokia"
    },
    {
      brandname : "Realme"
    },
    {
      brandname : "Mi"
    }
  ]);

  return (
    <store.Provider value={[data, setData]}>
      <center>
        <Count />
        <Display />
      </center>
    </store.Provider>
  )
}

export default App
```

Count.js

```
import React, {useContext} from 'react'
import {store} from './App'

const Count = () => {
  const [data, setData] = useContext(store);

  return (
    <div className='card' style={{'maxWidth': "50%", 'margin': "10px"}>
      <div className='card-body'>
        <h3 className='card-title'>Count : {data.length}</h3>
      </div>
    </div>
  )
}

export default Count
```

Display.js

```
import React, {useContext, useState} from 'react'
import {store} from './App'

const Display = () => {
  const [data, setData] = useContext(store);
  const [name, setName] = useState('');
  const submitHandler = e =>{
    e.preventDefault();
    setData([...data, {brandname: name}])
  }
}
```

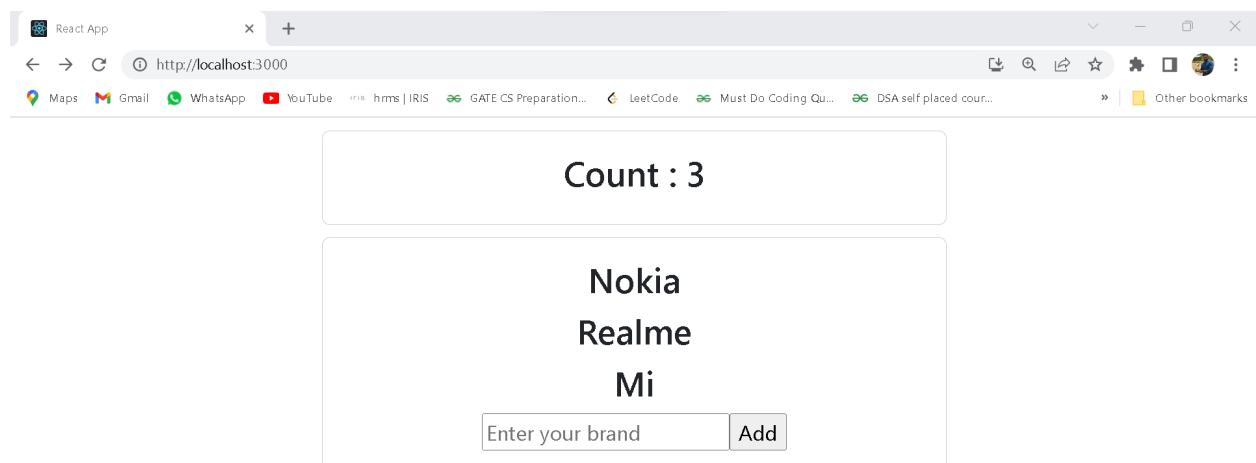
```

    return (
      <div className='card' style={{ "maxWidth": "50%" ,
"margin": "10px" }}>
        <div className='card-body'>
          {data.map(item => <h3
            className='card-title'>{item.brandname}</h3>) }
          <form className='form' onSubmit={submitHandler}>
            <input type='text' onChange={(e)
=>setName(e.target.value)} placeholder='Enter your brand' />
            <input type='submit' value='Add' />
          </form>
        </div>
      </div>
    )
}

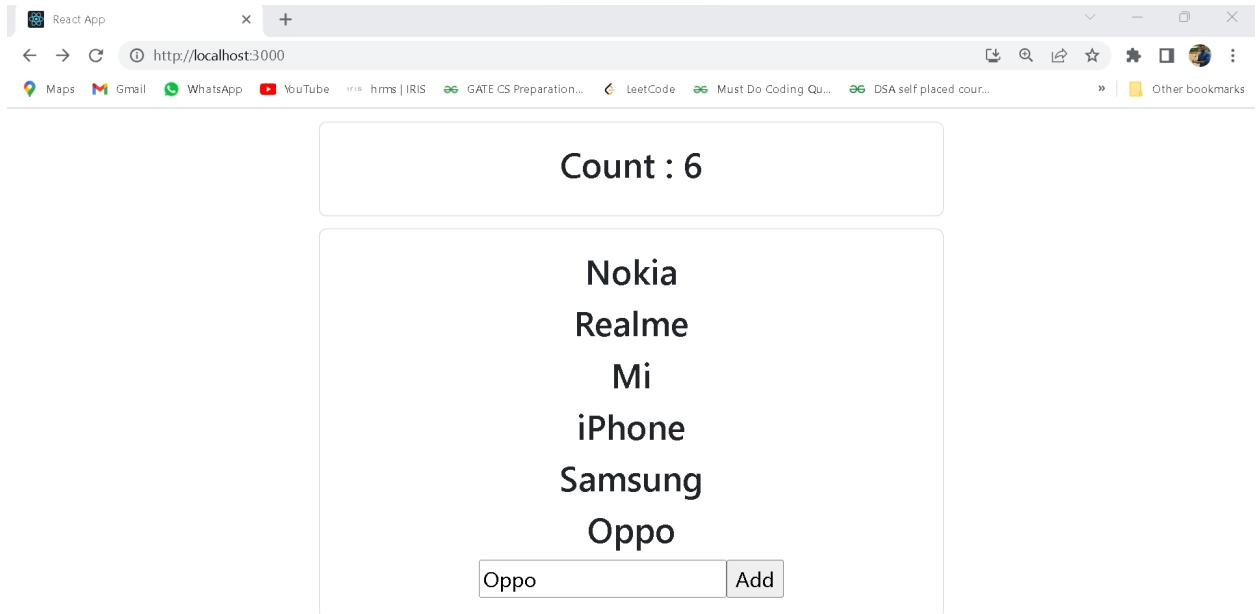
export default Display

```

O/P:



→After Adding items



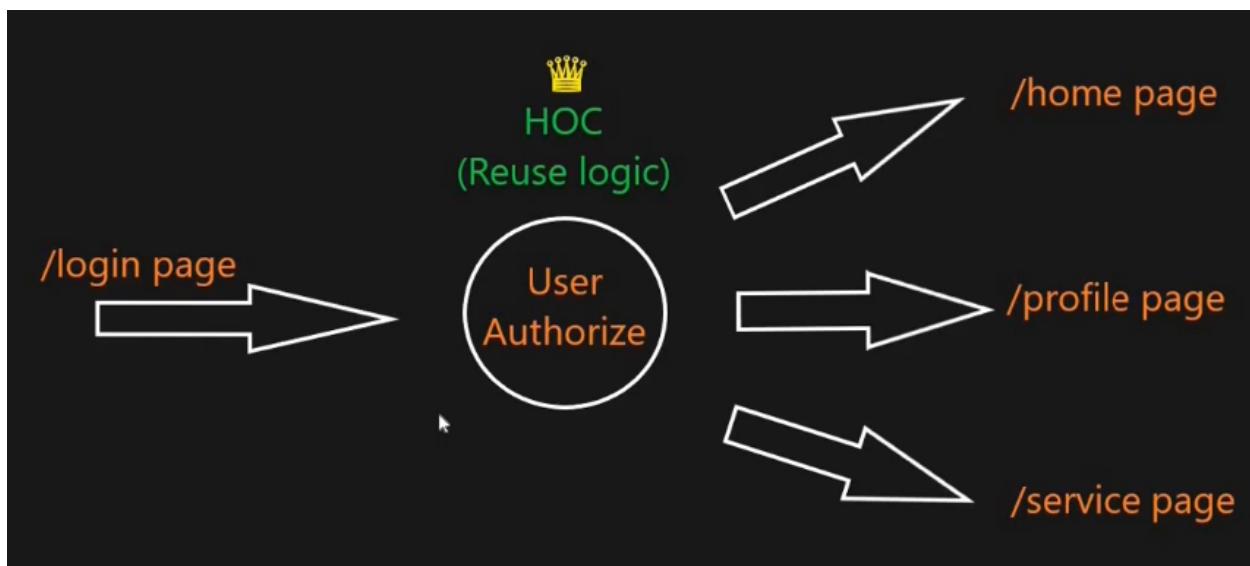
Higher Order Component in ReactJs

What is HOC?

Higher-order component is a function that **takes a component and returns a new component.**

→Advantages of HOC

- A higher-order component (HOC) is an advanced technique in React for reusing component logic.
- HOC is a pure function with zero side-effects and doesn't modify the input component.



App.js

```
import React from 'react'  
import HOC from './HOC'
```

```
const App = (props) => {
  return (
    <h1>Welcome User !! {props.name} </h1>
  )
}

export default HOC(App) //wrapping with HOC
```

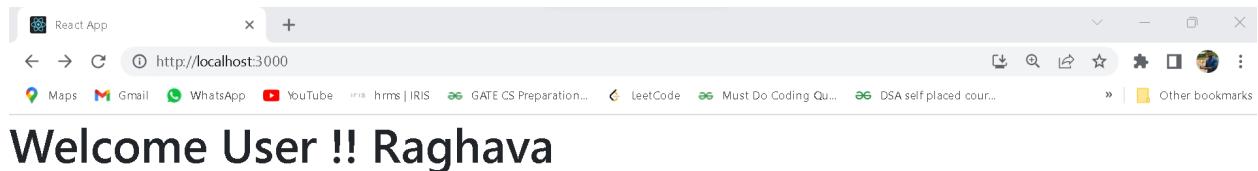
HOC.js

```
import React from 'react'

const HOC = (Component) => {
  return (
    class extends React.Component {
      state = {
        auth : true
      }
      render() {
        return(
          <div>
            {this.state.auth ? <Component
name="Raghava" /> : <h1>Please Login</h1>}
          </div>
        )
      }
    }
  )
}

export default HOC
```

O/P:



useMemo hooks in ReactJs

→ It needs to increase the performance of entire Application

→ **Definition**

- Help us to **improve** the performance of application, when we are performing most **expensive** function

- useMemo will **not run** for every re-render happens

- It will **run** during the first render and when it's **dependency** values changes.

Difference b/w useEffect() & useMemo()?

- useEffect() will run after render of the component.
- useMemo() will run during render of the component.

App.js

```
import React from 'react'

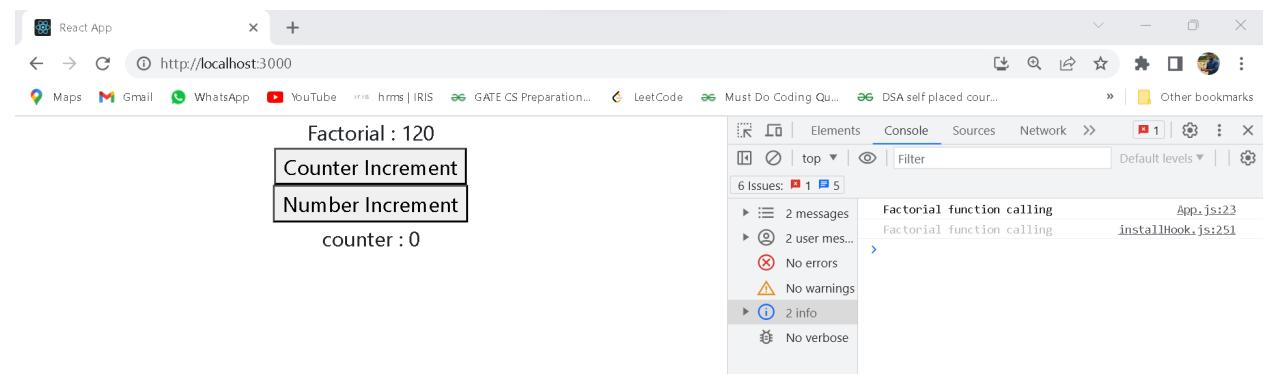
const App = () => {
  const [counter, setCounter] = React.useState(0);
  const [number, setNumber] = React.useState(5);
  const Factorial = React.useMemo(()=>{fact(number)}, [number])

  return (
    <center>
      Factorial : {Factorial} <br />
      <button onClick={() => setCounter(counter+1)}>Counter
      Increment</button> <br />
      <button onClick={() => setNumber(Number+1)}>Number
      Increment</button> <br />
      counter : {counter}
    </center>
  )
}
```

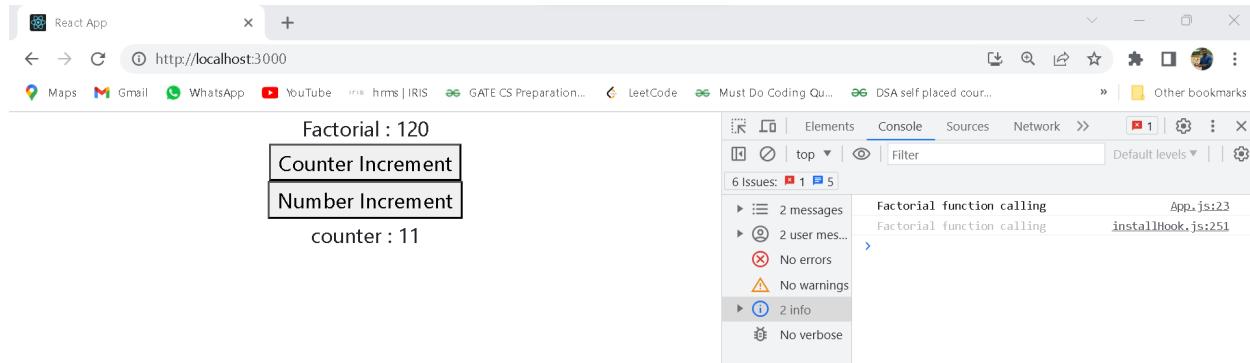
```
)  
}  
  
const fact = (n) =>{  
    let answer = 1;  
    for ( var i = n; i >=1; i--) {  
        answer = answer * i;  
    }  
    console.log('Factorial function calling');  
    return answer;  
}  
  
export default App
```

O/P:

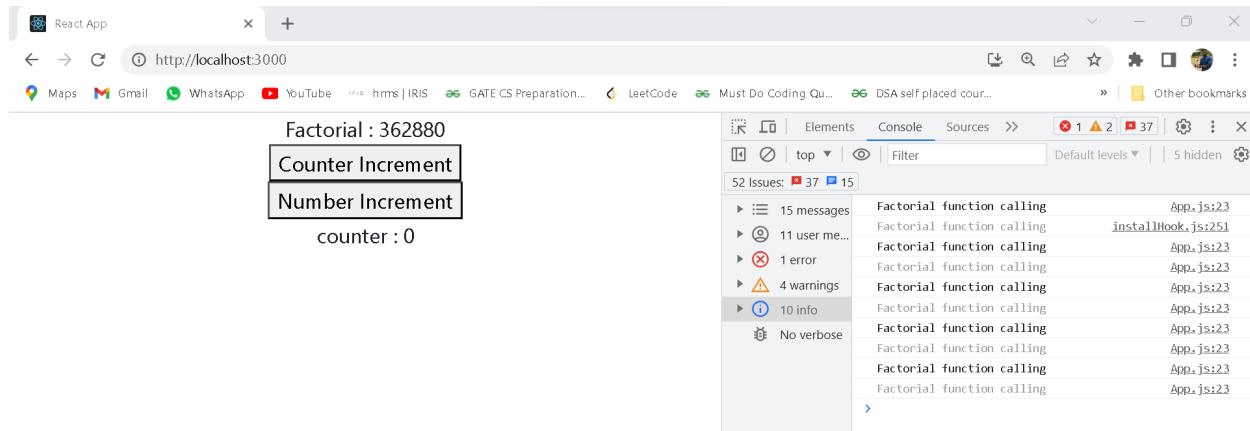
→Initial



→ Increment counter then factorial function is not calling because of **useMemo()**



→ Increment number Factorial is calling



useMemo vs useEffect in ReactJs

```
useEffect ( () => {} , [a,...] )
useMemo ( () => {} , [a,..] )
```

App.js

```
import React from 'react'

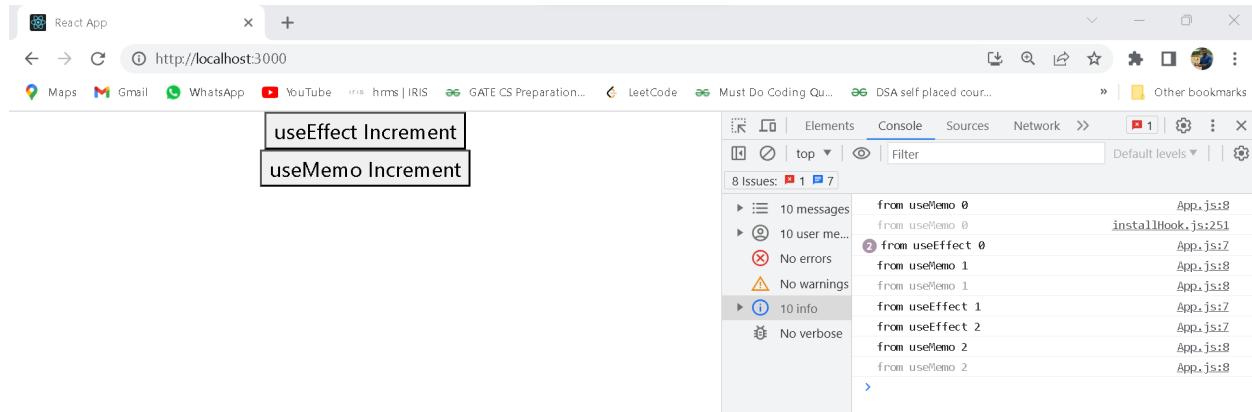
const App = () => {
  const [effect, setEffect] = React.useState(0);
  const [memo, setMemo] = React.useState(0);

  React.useEffect(()=>{console.log('from useEffect ' +
effect)}, [effect]);
  React.useMemo(()=>{console.log('from useMemo ' + memo)}, [memo]);

  return (
    <center>
      <button onClick={() => setEffect(effect+1)}>useEffect Increment</button> <br />
      <button onClick={() => setMemo(memo+1)}>useMemo Increment</button> <br />
    </center>
  )
}

export default App
```

O/P:



Stateful and Stateless Components in ReactJS

App.js

→ Stateful

```
import React from 'react'
import Test from './Test';

const App = () => {

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

  return (
    <center>
      <h3>Count : {count}</h3>
      <button onClick={() => setCount(count+1)}>
        Increment</button>
      <Test count= {count} />
    </center>
  )
}
```

```
}
```



```
export default App
```

Test.js

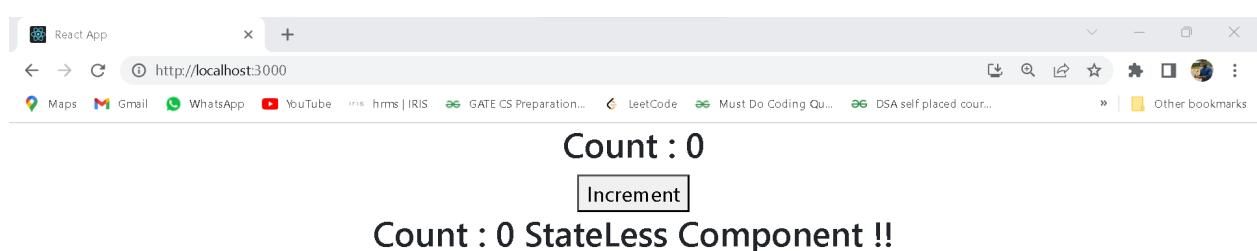
→StateLess

```
import React from 'react'

const Test = (props) => {
  return (
    <div>
      <center>
        <h3>Count : {props.count} StateLess Component
      ! !</h3>
        </center>
      </div>
  )
}

export default Test
```

O/P:



→After Increment



Controlled & UnControlled Components in ReactJs

Control Component :

Element data can be controlled by parent component through callbacks like `onChange()`

Uncontrolled Components :

Element's data can be controlled by the DOM itself

Control Component

App.js

```
import React from 'react'
import Controlled from './Controlled'
import Uncontrolled from './Uncontrolled'

const App = () => {
  return (
    <div>
      <center>
        <Controlled />
      </center>
    </div>
  )
}

export default App
```

Controlled.js

```
import React from 'react'
import Test from './Test';

const Controlled = () => {

  const [name, setName] = React.useState('');
  const changeHandler = e => {
    setName(e.target.value);
  }
  return (
    <div>
      <input type="text" value={name} onChange={changeHandler}>
      <br/>
      {name}
    </div>
  )
}

export default Controlled
```

```
<div>
  <center>
    Name : {name} <br />
    <input type='text' onChange={changeHandler} /> <br />
  </center>
</div>
)
}

export default Controlled
```

Test.js

```
import React from 'react'

const Test = (props) => {
  return (
    <div>
      <center>
        <input type='text'
onChange={props.changeHandler}/>
      </center>
    </div>
  )
}

export default Test
```

O/P:

→First Box form App.js



→Second Box form Test.js (overriding Raghava)



Uncontrolled Component

App.js

```
import React from 'react'
import Uncontrolled from './Uncontrolled'

const App = () => {
  return (
    <div>
      <center>
        <Uncontrolled />
      </center>
    </div>
  )
}

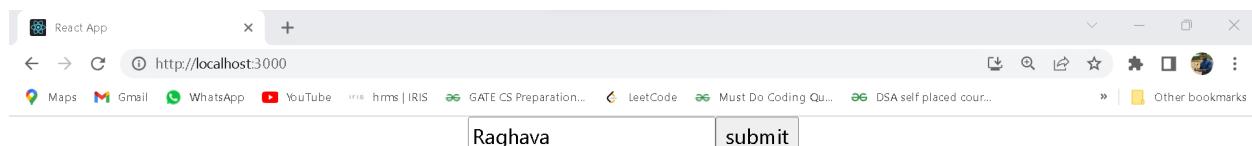
export default App
```

```
)  
}  
  
export default App
```

Uncontrolled.js

```
import React from 'react'  
  
function Example() {  
  const inputRef = React.useRef('');  
  const submitHandler = e => {  
    e.preventDefault();  
    alert(inputRef.current.value)  
  }  
  return (  
    <div>  
      <form onSubmit={submitHandler}>  
        <input type='text' ref= {inputRef} />  
        <input type='submit' value='submit' />  
      </form>  
    </div>  
  )  
}  
  
export default Example
```

O/P:



React Lazy Loading

Lazy loading used to increase the loading speed of application with dynamic importing concept.

<https://legacy.reactjs.org/docs/getting-started.html>

<https://legacy.reactjs.org/docs/code-splitting.html#gatsby-focus-wrapper>

Bundle → All files into one file. Like all .js into One Js & all .cssfiles into one file.

```
C:\Users\welcome\test3>npm run build
> test3@0.1.0 build
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

 49.38 kB (-56 B)  build\static\js\main.a7fbad5c.js
 1.78 kB           build\static\js\787.dfe8b844.chunk.js
 264 B            build\static\css\main.e6c13ad2.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

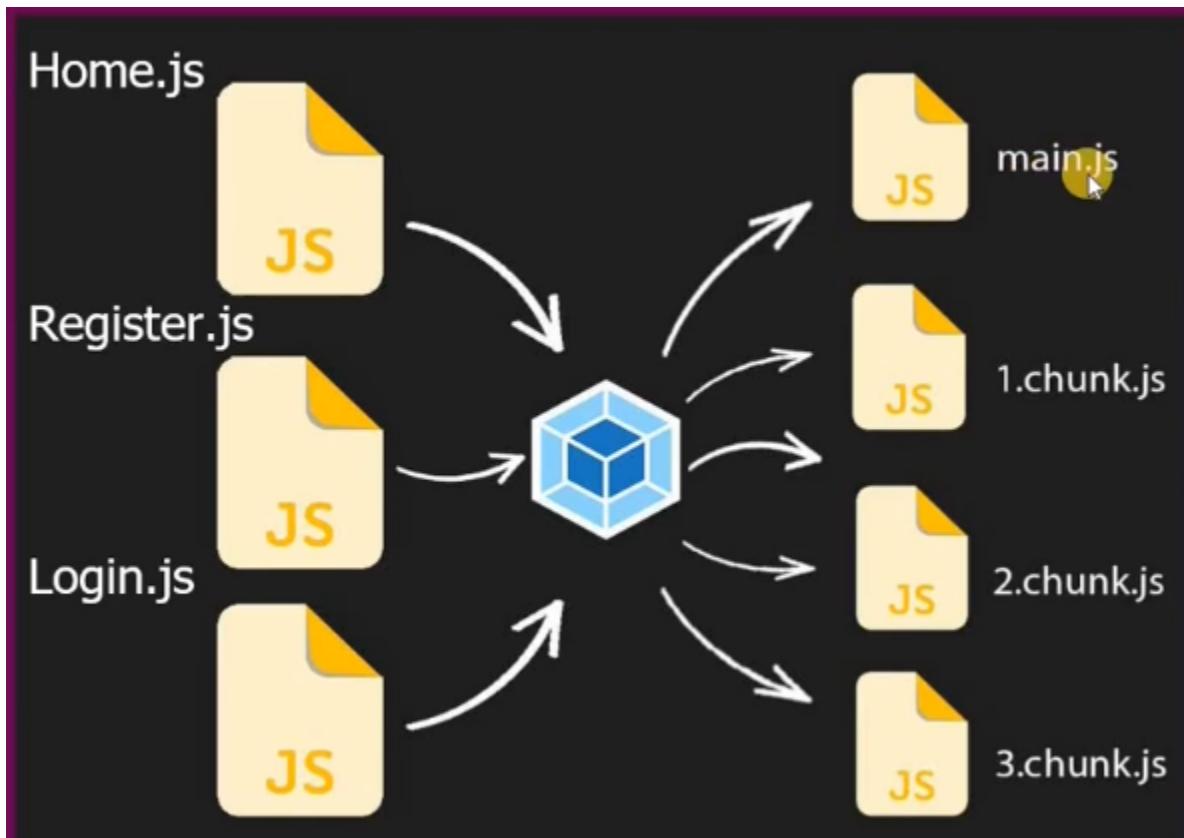
The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment
```

→Second time running the “npm run build” command



```
Select C:\Windows\System32\cmd.exe
> test3@0.1.0 build
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

 49.44 kB (+56 B)  build\static\js\main.7f770f0b.js
  1.78 kB          build\static\js\787.dfe8b844.chunk.js
  264 B            build\static\css\main.e6c13ad2.css
  217 B            build\static\js\461.97187ef6.chunk.js
  216 B            build\static\js\193.e19eafed.chunk.js
  212 B            build\static\js\66.13baee16.chunk.js

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment
```

```
src > JS App.js > ...
1 import React from 'react';
2 import { BrowserRouter, Routes, Route } from 'react-router-dom';
3 const Home = React.lazy(()=>import('./Components/Home'));
4 const Login = React.lazy(()=>import('./Components/Login'));
5 const Register = React.lazy(()=>import('./Components/Register'));
6
7 const App = () => {
8   return (
9     <BrowserRouter>
10    <Routes>
11      <Route path='/' element={<Home />}></Route>
12      <Route path='/login' element={<Login/>}></Route>
13      <Route path='/register' element={<Register/>}></Route>
14    </Routes>
15  )
16}
17
```

Error Boundary in ReactJs

→Generally We will use for Errors

→Imperative code in Try block

Try-catch block :

```
try {  
    ...  
}  
catch (err) {  
    ...  
}
```

→Error boundary (Declarative Code)

Error boundary component will catch JavaScript errors anywhere in their child component tree and display a fallback UI .

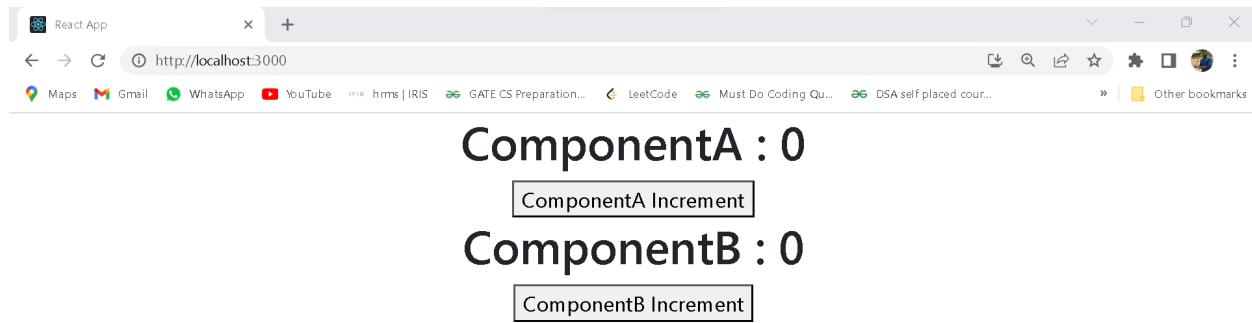
Ex:

```
<ErrorBoundary>
  <ComponentA /> -----> (child comp)
</ErrorBoundary>
```

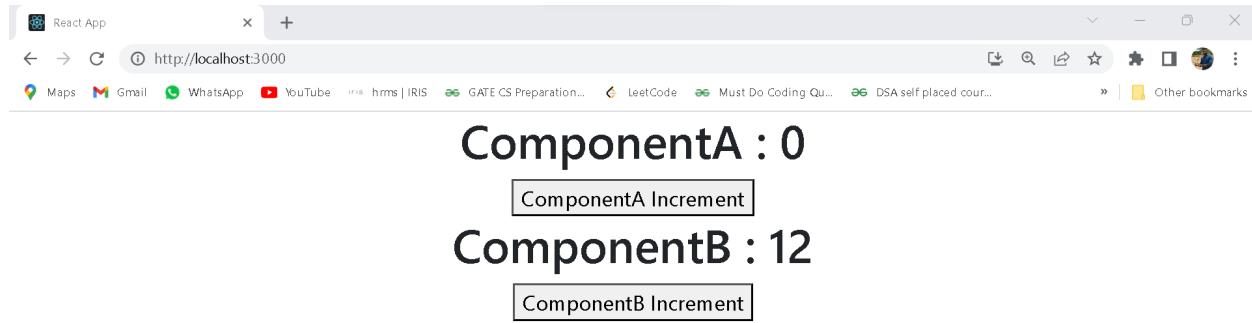


<ErrorBoundary> is not an react inbuilt importing component.

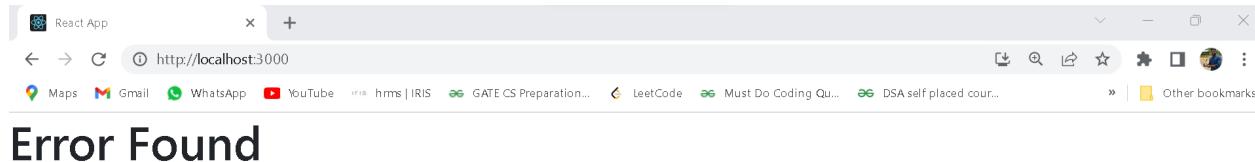
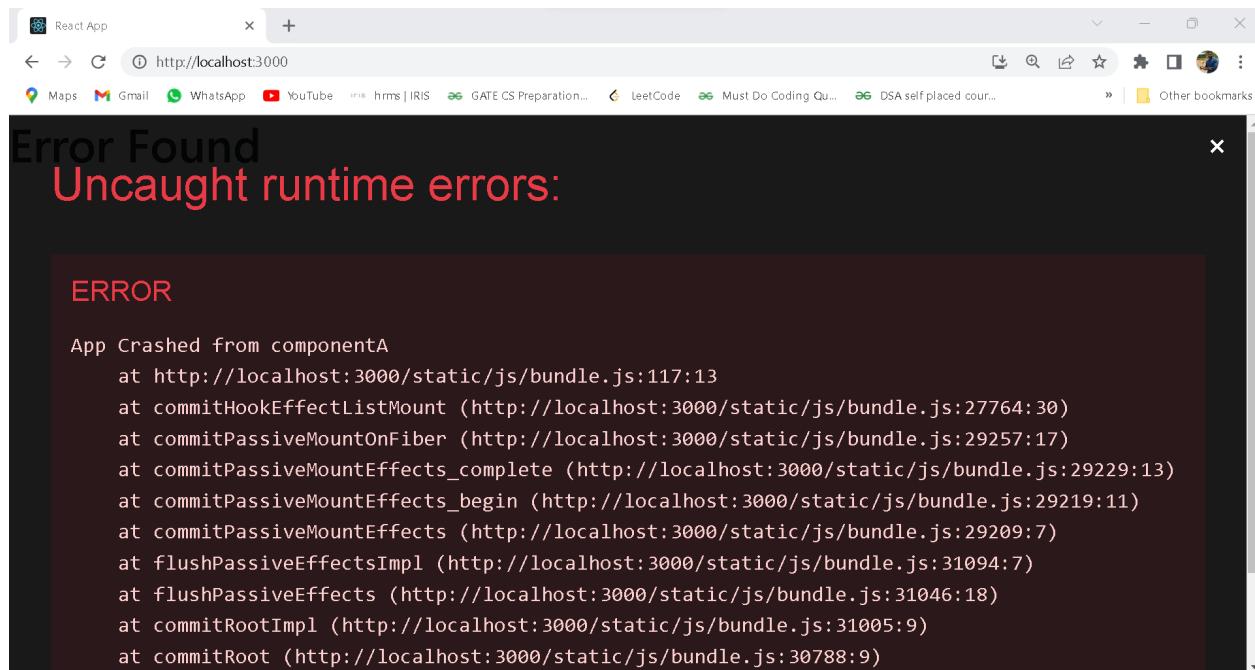
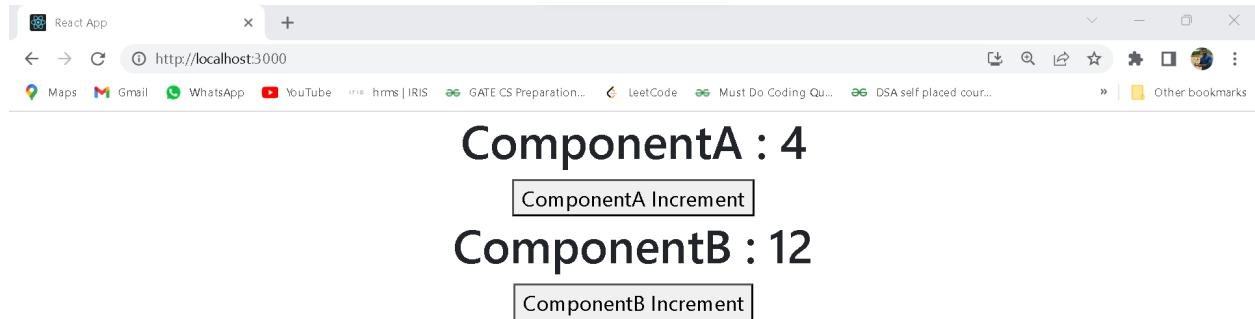
O/P:



→No Error in the ComponentB



→ In ComponentA when count ==5 it shows error



React Redux ToolKit

Redux toolkit is launched by Redux team, To solve three major issues with redux.

Configuring a Redux store is too complicated.

It required lot of packages to build a large scale application.

To reduce the complexity of boilerplate code.

Activate Windows
Go to Settings to activate Windows

#telugu

1. Store

```
const store = createStore(  
  reducer,  
  composeWithDevTools(applyMiddleware(...thunk))  
)
```

Redux - Flow

2. Reducers

```
export default function reducer(state = count,action){  
  const {type,payload} = action;  
  switch(type){  
    case "INCREMENT":  
      return state+1;  
    case "DECREMENT":  
      return state-1;
```

3. Actions

```
export const IncAction = (value) => async dispatch =>{  
  dispatch({  
    type:"INCREMENT",  
    payload: value  
  })
```

4. Using state values and Dispatch Actions

```
<button onClick={() => IncAction(5)} >Increment</button>  
<button onClick={DecAction} >Decrement</button>
```

Activate Windows
Go to Settings to activate Windows

Redux - Flow

1. Store

```
const store = createStore(  
  reducer,  
  composeWithDevTools(applyMiddleware(...thunk))  
)
```

-> createSlice

2. Reducers

```
export default function reducer(state = count,action){  
  const {type,payload} = action;  
  switch(type){  
    case "INCREMENT":  
      return state+1;  
    case "DECREMENT":  
      return state-1;
```

3. Actions

```
export const IncAction = (value) => async dispatch =>{  
  dispatch({  
    type:"INCREMENT",  
    payload: value  
  })
```

4. Using state values and Dispatch Actions

```
<button onClick={() => IncAction(5)} >Increment</button>  
<button onClick={DecAction} >Decrement</button>
```

Activate Windows

1. Store

Redux Toolkit - Flow

```
export const Store = configureStore({  
  reducer:{  
    reducers: reducer  
  }  
})
```

2. createSlice

```
const Count = createSlice({  
  name:"Count",  
  initialState,  
  reducers:{  
    increment:(state) =>{  
      state.count +=1  
    },  
    decrement : (State) => {  
      ...  
      state.count -= 1  
    }  
  }  
)
```

3. use state values and dispatch

```
<button onClick={() => dispatch(increment())}>Increment</button>  
<button onClick={() => dispatch(decrement())}>Decrement</button>
```

→ We need to develop Redux Toolkit required these packages

- npm install @reduxjs/toolkit react-redux
- npm install bootstrap

PS C:\Users\gudiw\Documents\tuts\sample> **npm install @reduxjs/toolkit
react-redux**

added 2 packages, and audited 1557 packages in 24s

244 packages are looking for funding
run `npm fund` for details

7 vulnerabilities (1 moderate, 6 high)

To address issues that do not require attention, run:
npm audit fix

fix --force

Run `npm audit` for details.

PS C:\Users\gudiw\Documents\tuts\sample> **npm install bootstrap**

up to date, audited 1557 packages in 3s

244 packages are looking for funding
run `npm fund` for details

7 vulnerabilities (1 moderate, 6 high)

To address issues that do not require attention, run:

```
npm audit fix
```

To address all issues (including breaking changes), run:

```
npm audit fix --force
```

Run `npm audit` for details.

→ They developed a small project.