React Js

1. How to create a react project

Asus@ASUS-VIVOBOOK-PRO-15 MINGW64 /d/IMPMyWorkplace/React (main)

**$ npx create-react-app NameOfProject**

1. how to run app

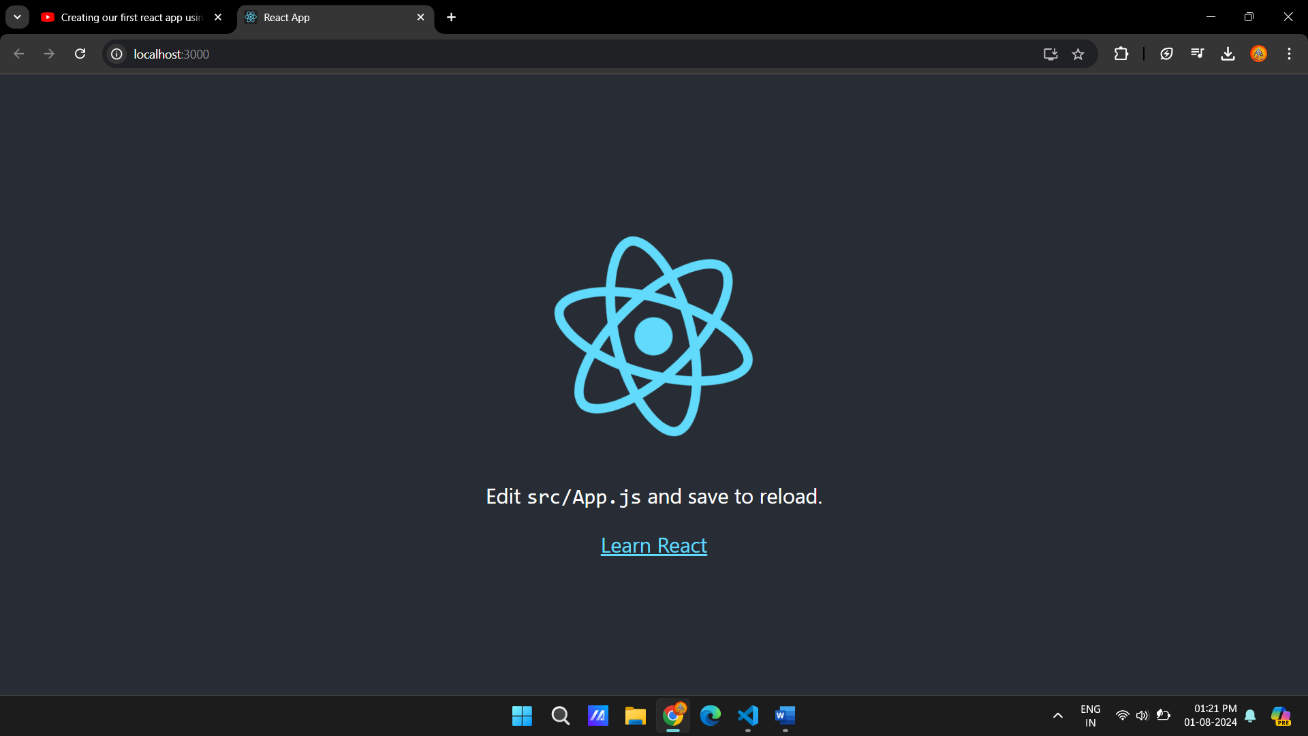
Asus@ASUS-VIVOBOOK-PRO-15 MINGW64 /d/IMPMyWorkplace/React (main)

**$ cd my-app1**

Asus@ASUS-VIVOBOOK-PRO-15 MINGW64 /d/IMPMyWorkplace/React/my-app1 (main)

**$ npm start**

1. output

****

1. basic javascript

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <script>

        // "use strict";

        m = 30;//if not strict mode fine

        // ReferenceError: m is not defined

        console.log('hi this is how printing')

        // this is comment

        var a = 20;

        var a = 40;

        console.log('this is how variable declared', a);// this is how variable declared 40

        const x = 10;

        //x=10+20;  //!TypeError: Assignment to constant variable. //means we can not update it

        let y = 10;

        // let y=30; //Cannot redeclare block-scoped variable 'y'.

        let z = {

            r: 20,

            a: 30,

            b: 50,

            c: function myfunction(k) {

                console.log(k);

            }

        }

        console.log(z);

        /\*

            index.html:31 {r: 20, a: 30, b: 50, c: ƒ}

            z.c(5)

            5

            z.a

            30

        \*/

        function myFun(name) {

            console.log('hi ', name);

        }

        /\*

            myFun()

            hi  undefined

        \*/

        document.addEventListener('click', function click() {

            console.log('this is printing when user click on h1');

            //alert('this is alert');

            /\*

                this is printing when user click on h1

                [Violation] 'click' handler took 1784ms

            \*/

            let conf = confirm('are sure?');

            console.log(conf);

        });

        /\*

            true/false

            'click' handler took 2306ms

        \*/

        arr = [10, 20, 30];

        console.log(arr);//[10, 20, 30]

        arr.forEach(element => {

            console.log(element);

        });

        /\*

            10

            20

            30

        \*/

        console.log(this);//Window {window: Window, self: Window, document: document, name: '', location: Location, …}

        function fun1() {

            console.log(this);

        }

        fun1();//Window {window: Window, self: Window, document: document, name: '', location: Location, …}

        fun2 = (number) => {

            console.log("the number is", +  this);

        }

        fun2();//the number is NaN

        setTimeout(()=>{

            console.log('I am callback function==> i will print after 3 milisecound');

        },3000);

        console.log('i am below of call back function');

        /\*

            i am below of call back function

            I am callback function==> i will print after 3 milisecound

        \*/

</script>

</head>

<body>

    <h1>Hi I am learning javascript</h1>

</body>

</html>

1. imp point

we can not directly add multiple tags in App.js

we got an error JSX expressions must have one parent element.

import './App.css';

function App() {

  return (

    <div className="1">

      <h1>1</h1>

    </div>

    <div className="2">

      <h1>2</h1>

    </div>

  );

}

export default App;

so, to do that we have an option

we simply write our code in <> </> in this braces

import './App.css';

function App() {

  return (

    <>

      <div className="1">

        <h1>1</h1>

      </div>

      <div className="2">

        <h1>2</h1>

      </div>

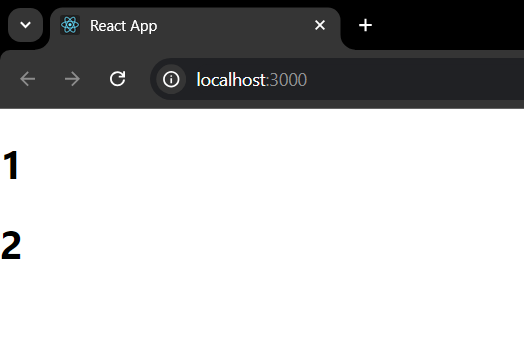
    </>

  );

}

export default App;

no error



1. is we can add js directly in App.js let’s try it

import './App.css';

let name = "Suraj";

function App() {

  return (

    <>

      <p>trying to print name variable <b>name</b></p>

      {/\* not printing variable name \*/}

      <p>trying to print name variable<b>{name}</b> </p>

      {/\* it will print variable name \*/}

    </>

  );

}

export default App;

so, to do that we simply use { } brackets to print variables from js to react

**let a=”Data”;**

**function App(){**

**return(**

**<h1> { a } </h1>**

**);**

**}**

1. now how to add bootstrap to our app

[**https://getbootstrap.com/docs/5.3/getting-started/introduction/**](https://getbootstrap.com/docs/5.3/getting-started/introduction/)

visit this link and then copy script and css and add it to index.html file in public folder

//my-app1\public\index.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8" />

  <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <meta name="theme-color" content="#000000" />

  <meta name="description" content="Web site created using create-react-app" />

  <link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

  <link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

  <title>React App</title>

  <!-- added bootstrap css  -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"

    integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

</head>

<body>

  <noscript>You need to enable JavaScript to run this app.</noscript>

  <div id="root"></div>

  <!-- addded bootstrap script  -->

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"

    integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"

    crossorigin="anonymous"></script>

</body>

</html>

1. now how to copy bootstrap component and add to project

<https://getbootstrap.com/docs/5.3/components/navbar/>

adding navbar to the App.js

* after copying component, we need to check is there any tag that not closed

like,

* hr tag
* img tag
* input tag

so, we need to manually close those tags or else getting error JSX element 'input' has no corresponding closing tag.

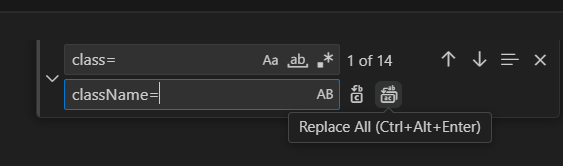
* One more thing is important like we need to change href=” #” to href=”/” because we need to make routes not simply change page or else we get an error

The href attribute is required for an anchor to be keyboard accessible. Provide a valid, navigable address as the href value. If you cannot provide an href, but still need the element to resemble a link, use a button and change it with appropriate styles.

* Now we can not use class directly In app.js because it will consider class as in js so we need to change it with className

**Class**🡺**className**

**To change cltr+f then add “class=” in search bar and replace all with “class Name”**

****

now our App.js looks like

import './App.css';

function App() {

  return (

    <>

      <nav className="navbar navbar-expand-lg bg-body-tertiary">

        <div className="container-fluid">

          <a className="navbar-brand" href="/">Navbar</a>

          <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

            <span className="navbar-toggler-icon"></span>

          </button>

          <div className="collapse navbar-collapse" id="navbarSupportedContent">

            <ul className="navbar-nav me-auto mb-2 mb-lg-0">

              <li className="nav-item">

                <a className="nav-link active" aria-current="page" href="/">Home</a>

              </li>

              <li className="nav-item">

                <a className="nav-link" href="/">About</a>

              </li>

            </ul>

            <form className="d-flex" role="search">

              <input className="form-control me-2" type="search" placeholder="Search" aria-label="Search" />

              <button className="btn btn-outline-success" type="submit">Search</button>

            </form>

          </div>

        </div>

      </nav>

    </>

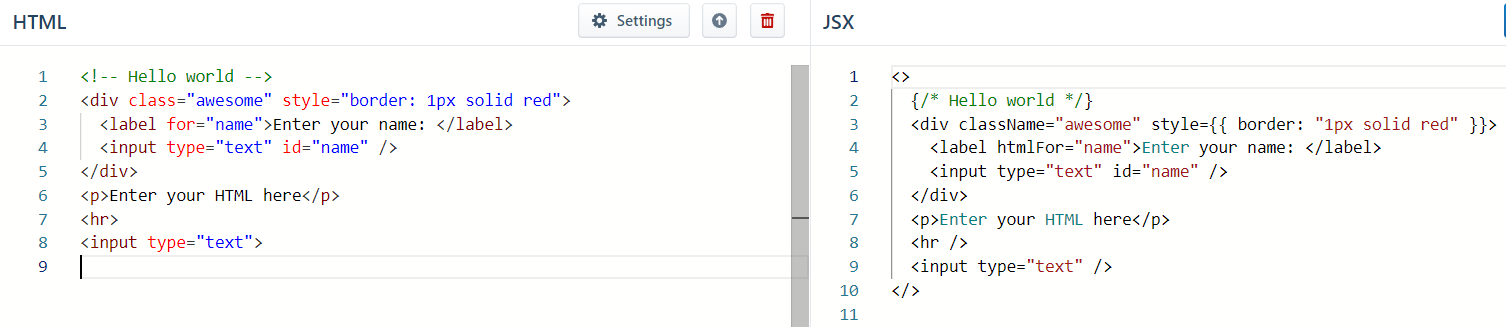
  );

}

export default App;

1. So why actually need to change

Because App.js uses JSX language which is similar to html



JSX looks like HTML, but under the hood it is transformed into plain JavaScript objects. You can’t return two objects from a function without wrapping them into an array. This explains why you also can’t return two JSX tags without wrapping them into another tag or a Fragment.

1. Modules in js **(to run default file open it in its directory node ./name.js)**

Named modules and default modules

* If we declared variable in one module and export it by default then while importing it to any other module, we can import it by using any name it will work

Module2.mjs =>

Const a=20;

Export default a;

Module1.mjs =>

Import xyz from ‘module2.mjs’

Console.log(xyz); //works

* But of we exporting it using export only(which means named export), then while importing we must define same name as present it in its own name

Module2.mjs =>

const a=20;

export a;

Module1.mjs =>

Import xyz from ‘module2.mjs’

Console.log(xyz); //won’t work

//my-app1\src\**module2.mjs**

const a="suraj";

const b="adarsh";

const c="shubh";

const d="sonya";

export default d;

export {a,b,c};

//my-app1\src\**module1.mjs**

import xyz,{a,b,c} from './module2.mjs'

// import xyz,{x1,x2,x3} from './module2.mjs'

//for named we must define same name as declared or while exporting and importing

//!import xyz, { x1, x2, x3 } from './module2.mjs'

//!              ^^

//!SyntaxError: The requested module './module2.mjs' does not provide an export named 'x1'

//this will not work

// console.log("a:"+x1);

// console.log("b:"+x2);

// console.log("c:"+x3);

//console.log("d:"+xyz);

console.log("a:"+a);

console.log("b:"+b);

console.log("c:"+c);

console.log("d:"+xyz);

output:

**React/my-app1 (main)**

**$ node ./src/module1.mjs**

**a:suraj**

**b:adarsh**

**c:shubh**

**d:sonya**

1. What is component and how to use it in js

The above task we can done with component means adding code in one file an using it in App.js

Using class name of that component

import './App.css';

import Navbar from './components/Navbar';//importing component

function App() {

  return (

    <>

//adding name of a component

      <Navbar />

    </>

  );

}

export default App;

//my-app1\src\components\Navbar.js

import React from 'react'; //imr

//rfc ==> react function based component

export default function Navbar() {

    return (

        <nav className="navbar navbar-expand-lg bg-body-tertiary">

            <div className="container-fluid">

                <a className="navbar-brand" href="/">Navbar</a>

                <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

                    <span className="navbar-toggler-icon"></span>

                </button>

                <div className="collapse navbar-collapse" id="navbarSupportedContent">

                    <ul className="navbar-nav me-auto mb-2 mb-lg-0">

                        <li className="nav-item">

                            <a className="nav-link active" aria-current="page" href="/">Home</a>

                        </li>

                        <li className="nav-item">

                            <a className="nav-link" href="/">About</a>

                        </li>

                    </ul>

                    <form className="d-flex" role="search">

                        <input className="form-control me-2" type="search" placeholder="Search" aria-label="Search" />

                        <button className="btn btn-outline-success" type="submit">Search</button>

                    </form>

                </div>

            </div>

        </nav>

    )

}

1. What is prop in js

Prop basically a property for short which are basically used to pass arguments from functions

In a component if we need a component but by its different property in above ex if I want navbar but with different title name then that time prop will be used.

import React from 'react'; //imr

import PropTypes from 'prop-types'; //impt

//rfc ==> react function based component

export default function Navbar(prop) {

    return (

        <nav className="navbar navbar-expand-lg bg-body-tertiary">

            <div className="container-fluid">

//here defining a prop for title

                <a className="navbar-brand" href="/">{prop.title}</a>

                <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

                    <span className="navbar-toggler-icon"></span>

                </button>

                <div className="collapse navbar-collapse" id="navbarSupportedContent">

                    <ul className="navbar-nav me-auto mb-2 mb-lg-0">

                        <li className="nav-item">

//here defining a prop for title

                            <a className="nav-link active" aria-current="page" href="/">{prop.home}</a>

                        </li>

                        <li className="nav-item">

//here defining a prop for title

                            <a className="nav-link" href="/">{prop.about}</a>

                        </li>

                    </ul>

                    <form className="d-flex" role="search">

                        <input className="form-control me-2" type="search" placeholder="Search" aria-label="Search" />

                        <button className="btn btn-outline-success" type="submit">Search</button>

                    </form>

                </div>

            </div>

        </nav>

    )

}

   import './App.css';

      import Navbar from './components/Navbar';//

      function App() {

        return (

          <>

            <Navbar title="Suraj" home="Home" about="About"/>

          </>

        );

      }

      export default App;

but what if we forgot to add prop in app.js then it will not show that data. So, for that we can define its type and requirement

import React from 'react'; //imr

import PropTypes from 'prop-types'; //impt

//rfc ==> react function based component

export default function Navbar(prop) {

    return (

        //code

//code

    )

}

Navbar.propTypes =

{

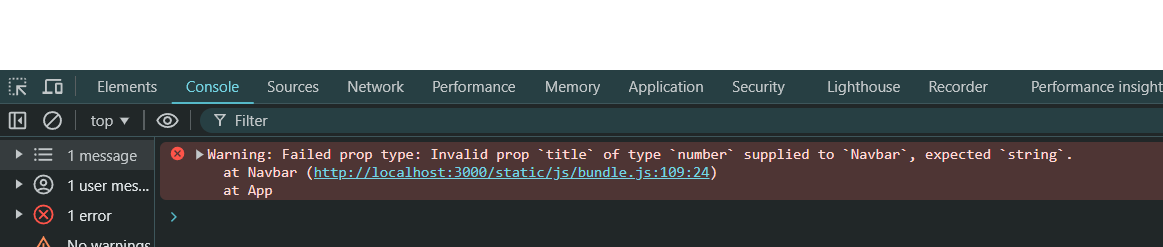
    title: PropTypes.string, //pts

    home: PropTypes.string,

    about: PropTypes.string,

}

And now if we trying to add number instead of String then we got an error



**Now how to make them required**

Navbar.propTypes =

{

    title: PropTypes.string.isRequired,//pts

    home: PropTypes.string.isRequired,

    about: PropTypes.string.isRequired,

}

1. State in react and how to handle it?

//how to tell react that we are going to use a useState

import React, { useState } from 'react'

**React hooks are functions that allow functional components to tap into state and lifecycle features previously exclusive to class components, making it easier to manage stateful logic in a more consistent way**

Because we are using function component, we called useState so it is a hook.

//this is how we define a initial state and function to update it

// Returns a stateful value, and a function to update it.

    const [text, setText] = useState("Enter Text Here");

* text holds the **current state** (initially set to "Enter Text Here").
* setText is the **function** to update the text state.

So question we to change state

setText="hello"; incorrect way to handel state

setText("hello"); correct way to handel state

1. create a component

//my-app1\src\components\TextForm.js

// this is how we using hooks in react(useState is called hook beacause of we are try to using it in a functional base component)

import React, { useState } from 'react'

export default function TextForm(prop) {

    function onClickUpper() {

        let newText = text.toUpperCase();

        setText(newText);

    }

    function onClickLower() {

        let newText = text.toLowerCase();

        setText(newText);

    }

    //in text area always gives us a event

    function handelOnChange(event) {

        // setText="hello"; incorrect way to handel state

        // setText("hello"); correct way to handel state

        //we can change what user is typing in textarea

        setText(event.target.value);

    };

    // Returns a stateful value, and a function to update it.

    const [text, setText] = useState("Enter Text Here");

    return (

        <div>

            <div className="mb-3">

                <textarea className="form-control" id="exampleFormControlTextarea1" value={text} onChange={handelOnChange} rows="8"></textarea>

            </div>

            <button type="button" className="btn btn-primary" onClick={onClickUpper}>UpperCase</button>

            <button type="button" className="btn btn-primary" onClick={onClickLower}>LowerCase</button>

        </div>

    )

}

setText(event.target.value);

this is how we can get user typed text in text area if we remove it we can not type in text area

1. import this and use it in App.js

import './App.css';

import Navbar from './components/Navbar';//

import TextForm from './components/TextForm';

function App() {

  return (

    <>

      <Navbar title='TextUtils' home="Home" about="About" />

      <div className="container">

        <h3>Enter Text Below</h3>

        <TextForm />

      </div>

    </>

  );

}

export default App;

1. Output

Success

1. Using state adding word count and character length preview.

React, { useState } from 'react'

export default function TextForm(prop) {

    function onClickUpper() {

        let newText = text.toUpperCase();

        setText(newText);

    }

    function onClickLower() {

        let newText = text.toLowerCase();

        setText(newText);

    }

    function handelOnChange(event) {

        setText(event.target.value);

    };

    const [text, setText] = useState("");

    return (

        <>

        <div className='container py-2'>

            <div className="mb-3">

                <textarea className="form-control" id="exampleFormControlTextarea1" value={text} onChange={handelOnChange} rows="8"></textarea>

            </div>

            <button type="button" className="btn btn-primary mx-2" onClick={onClickUpper}>UpperCase</button>

            <button type="button" className="btn btn-primary mx-2" onClick={onClickLower}>LowerCase</button>

        </div>

        <div className="container">

            <h3>Summery</h3>

            <p>{text.split(" ").length} words and {text.length} characters</p>

            <h3>preview</h3>

            <p>{text.length===0?"please enter text":text}</p>

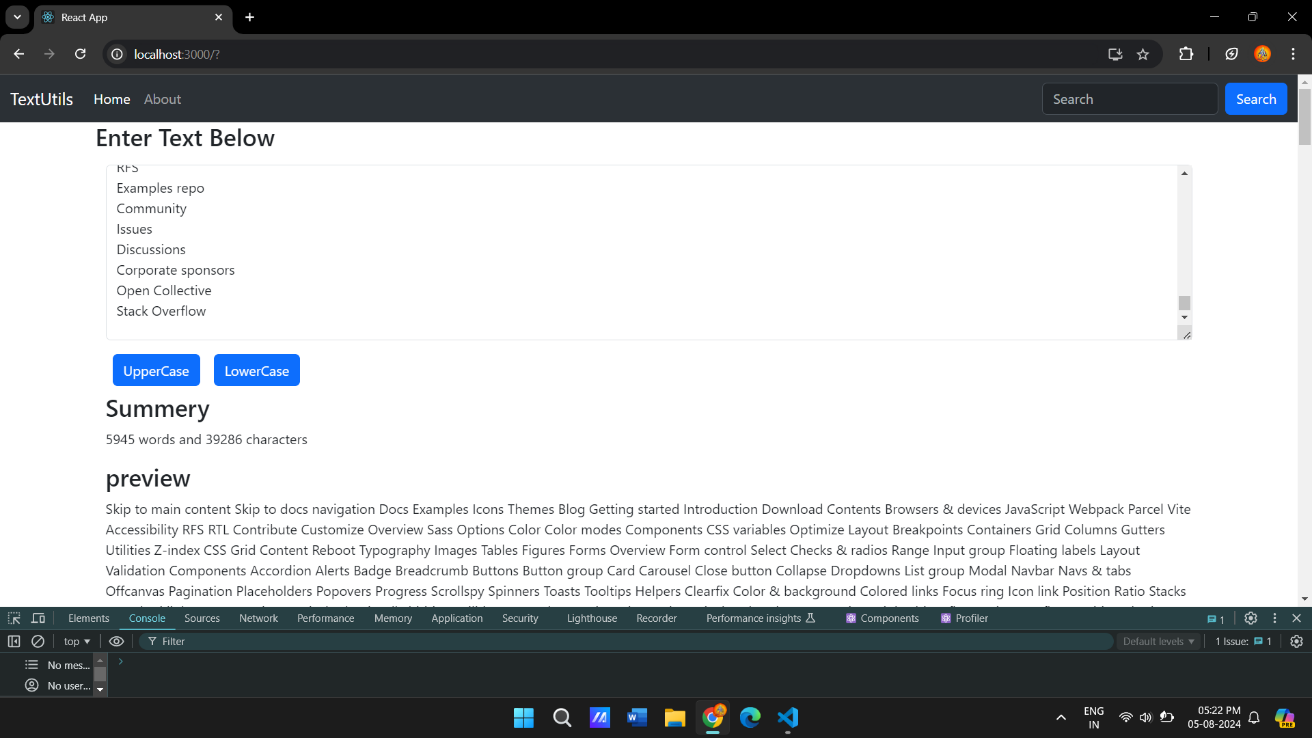
        </div>

        </>

    )

}

Output:



1. Toggle Theme using state
2. Create component

import React, { useState } from "react"

export default function About() {

    const [btnText, setBtnText] = useState("Enable Dark Mode");

    const [myStyle, setMyStyling] = useState({

        color: 'black',

        backgroundColor: 'white',

    });

    function toggleTheme() {

        if (myStyle.color === 'black') {

            setMyStyling(

                {

                    color: 'white',

                    backgroundColor: 'black',

                }

            );

            setBtnText('Enable light Mode');

        }

        else {

            setMyStyling({

                color: 'black',

                backgroundColor: 'white',

            });

            setBtnText('Enable dark Mode');

        }

    }

    return (

        <div style={myStyle}>

            <div className="accordion" id="accordionExample" style={myStyle}>

                <div className="accordion-item">

                    <h2 className="accordion-header">

                        <button className="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne" style={myStyle}>

                            Accordion Item #1

                        </button>

                    </h2>

                    <div id="collapseOne" className="accordion-collapse collapse show" data-bs-parent="#accordionExample">

                        <div className="accordion-body" style={myStyle}>

                            <strong>This is the first item's </strong>

                        </div>

                    </div>

                </div>

                <div className="accordion-item">

                    <h2 className="accordion-header">

                        <button className="accordion-button collapsed" style={myStyle} type="button" data-bs-toggle="collapse" data-bs-target="#collapseTwo" aria-expanded="false" aria-controls="collapseTwo">

                            Accordion Item #2

                        </button>

                    </h2>

                    <div id="collapseTwo" className="accordion-collapse collapse" data-bs-parent="#accordionExample">

                        <div className="accordion-body" style={myStyle}>

                            <strong>This is the second item's accordion body.</strong>

                        </div>

                    </div>

                </div>

                <div className="accordion-item">

                    <h2 className="accordion-header">

                        <button className="accordion-button collapsed" style={myStyle} type="button" data-bs-toggle="collapse" data-bs-target="#collapseThree" aria-expanded="false" aria-controls="collapseThree">

                            Accordion Item #3

                        </button>

                    </h2>

                    <div id="collapseThree" className="accordion-collapse collapse" data-bs-parent="#accordionExample">

                        <div className="accordion-body" style={myStyle}>

                            <strong>This is the third item's accordion body.</strong>

                        </div>

                    </div>

                </div>

            </div>

            <button type="button" onClick={toggleTheme} className="btn btn-primary my-3">{btnText}</button>

        </div>

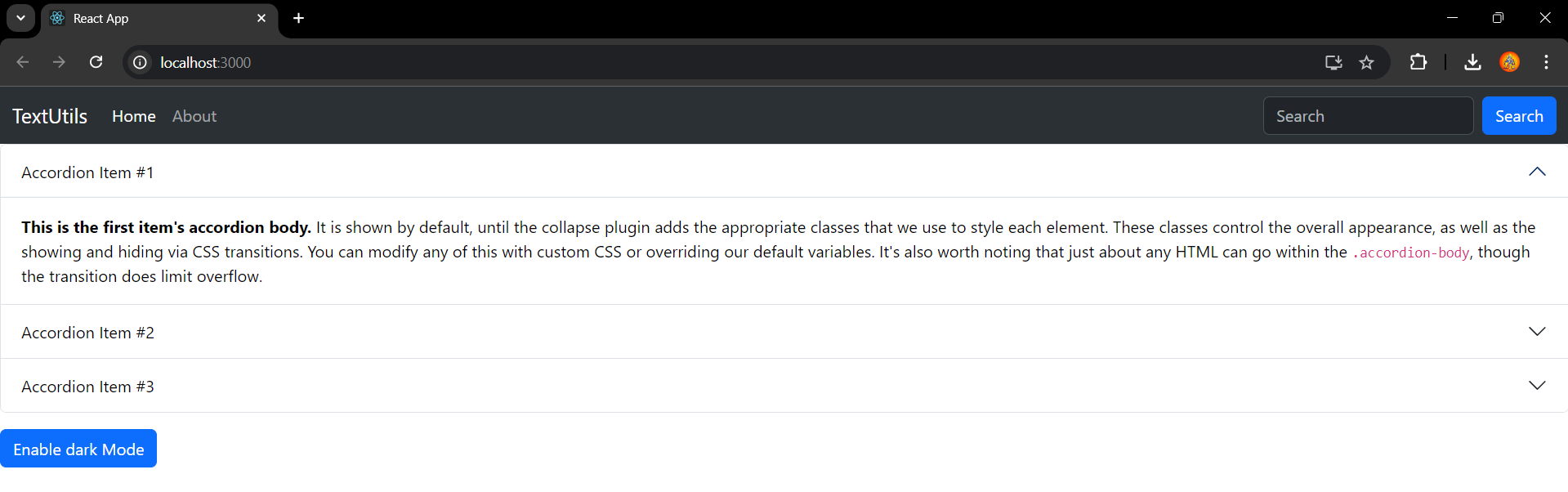
    )

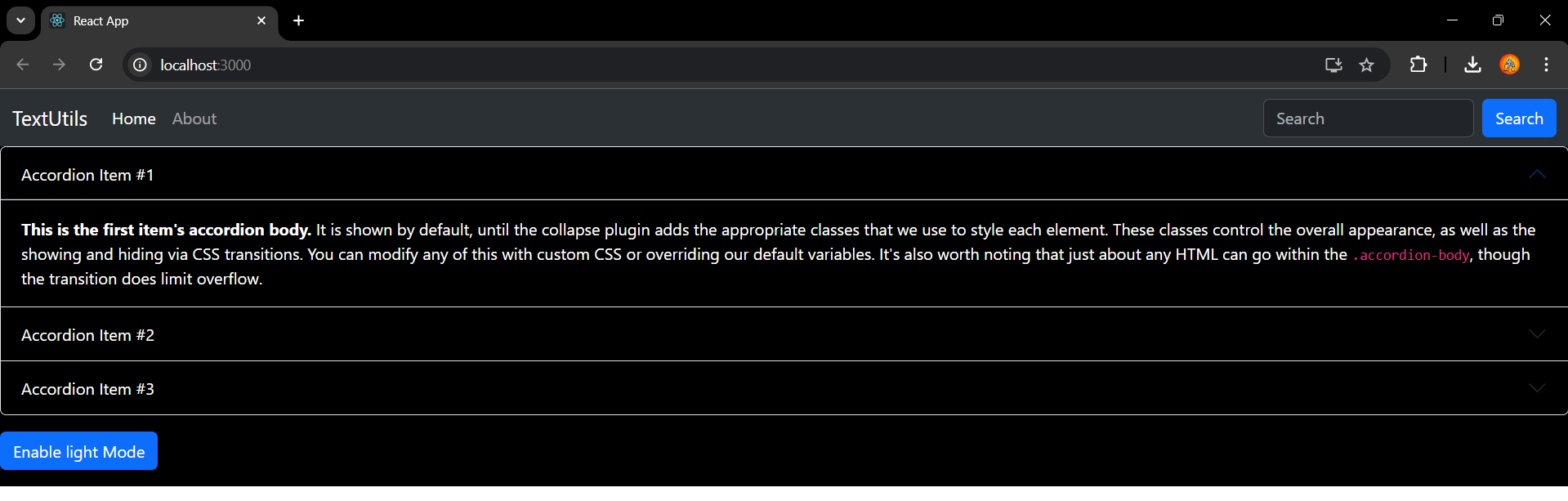
}

1. Add or import it in App.js

      <About />

1. Output





1. How to add copy and remove extra space in a text utils

import React, { useState } from 'react'

export default function TextForm(prop) {

//copy text

    const [copy, setCopy] = useState("Copy");

    function onClickCopy() {

        let newText = document.getElementById('exampleTextarea1');

        // newText.select();

        copy === 'Copy' ? setCopy("Copied") : setCopy("Copy");

        navigator.clipboard.writeText(newText.value);

    }

    //remove extra space

    function onClickRemoveExtraSpace() {

        //using regix with space>2

        //Split a string into substrings using the specified separator and return them as an array.

        let newText = text.split(/[  ]+/);

        //dds all the elements of an array into a string, separated by the specified separator string.

        setText(newText.join(" "));

    }

 return (

   <>

//code

<button type="button" className="btn btn-primary mx-2" onClick={onClickCopy}>{copy} Text</button>

                <button type="button" className="btn btn-primary mx-2" onClick={onClickRemoveExtraSpace}>Remove Extra Space</button>

  </>

    )

}

1. Now How to set a style from app.js also changing mode dark/light
2. In app.js make a state and function to toggle theme

import './App.css';

import About from './components/About';

import Navbar from './components/Navbar';

import TextForm from './components/TextForm';

import React, { useState } from 'react'//imrs

function App() {

  //set light initialy

  const [theme, setTheme] = useState('light');

  //fun to toogle theme

  function mode() {

    if (theme === 'light') {

      setTheme('dark');

      //universal setting

      document.body.style.backgroundColor = '#041630'

      document.body.style.color = 'white'

    }

    else {

      setTheme('light');

      //universal setting

      document.body.style.backgroundColor = 'white'

      document.body.style.color = 'black'

    }

  }

  return (

    <>

      <Navbar title='TextUtils' home="Home" about="About" theme={theme} mode={mode} />

      <TextForm theme={theme}/>

      {/\* <About /> \*/}

    </>

  );

}

export default App;

1. Now accept prop in pages and set style

import React, { useState } from 'react'

export default function TextForm(prop) {

    //in text area always gives us a event

    function handelOnChange(event) {

        setText(event.target.value);

    };

    // Returns a stateful value, and a function to update it.

    const [text, setText] = useState("");

    //upper case

    function onClickUpper() {

        let newText = text.toUpperCase();

        setText(newText);

    }

    //lower case

    function onClickLower() {

        let newText = text.toLowerCase();

        setText(newText);

    }

    //clear text

    function onClickClear() {

        setText('');

    }

    //copy text

    const [copy, setCopy] = useState("Copy");

    function onClickCopy() {

        let newText = document.getElementById('exampleTextarea1');

        // newText.select();

        copy === 'Copy' ? setCopy("Copied") : setCopy("Copy");

        navigator.clipboard.writeText(newText.value);

    }

    //remove extra space

    function onClickRemoveExtraSpace() {

        let newText = text.split(/[  ]+/);

        setText(newText.join(" "));

    }

    return (

        <>

            <div className="container">

                <h3>Enter Text Below</h3>

                <div>

                    <div className="mb-3">

                        <textarea className="form-control" id="exampleTextarea1" style={{ color: prop.theme === 'light' ? 'black' : 'white', backgroundColor: prop.theme === 'dark' ? 'grey' : 'white' }} value={text} onChange={handelOnChange} rows="8"></textarea>

                    </div>

                    <button type="button" className="btn btn-primary mx-2" onClick={onClickUpper}>UpperCase</button>

                    <button type="button" className="btn btn-primary mx-2" onClick={onClickLower}>LowerCase</button>

                    <button type="button" className="btn btn-primary mx-2" onClick={onClickClear}>Clear</button>

                    <button type="button" className="btn btn-primary mx-2" onClick={onClickCopy}>{copy} Text</button>

                    <button type="button" className="btn btn-primary mx-2" onClick={onClickRemoveExtraSpace}>Remove Extra Space</button>

                </div>

                <div className="container">

                    <h3>Summery</h3>

                    <p>{text.split(" ").length - 1} words and {text.length} characters</p>

                    <h3>preview</h3>

                    <p>{text.length === 0 ? "please enter text" : text}</p>

                </div>

            </div>

        </>

    )

}

import React from 'react'; //imr

import PropTypes from 'prop-types'; //impt

//rfc ==> react function based component

export default function Navbar(prop) {

    return (

        <nav className={`navbar navbar-expand-lg bg-body-tertiary bg-${prop.theme === 'light' ? 'dark' : 'light'}`} data-bs-theme={prop.theme}>

            <div className="container-fluid">

                <a className="navbar-brand" href="/">{prop.title}</a>

                <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

                    <span className="navbar-toggler-icon"></span>

                </button>

                <div className="collapse navbar-collapse" id="navbarSupportedContent">

                    <ul className="navbar-nav me-auto mb-2 mb-lg-0">

                        <li className="nav-item">

                            <a className="nav-link active" aria-current="page" href="/">{prop.home}</a>

                        </li>

                        <li className="nav-item">

                            <a className="nav-link" href="/">{prop.about}</a>

                        </li>

                    </ul>

                    <div className="form-check form-switch">

                        <input className="form-check-input" onClick={prop.mode} type="checkbox" role="switch" id="flexSwitchCheckDefault" />

                        <label className={`form-check-label text-${prop.theme === 'light' ? 'dark' : 'light'}`} htmlFor="flexSwitchCheckDefault" >{prop.theme === 'light' ? 'Enable DarkMode' :'Enable LightMode'}</label>

                    </div>

                </div>

            </div>

        </nav>

    )

}

Navbar.propTypes =

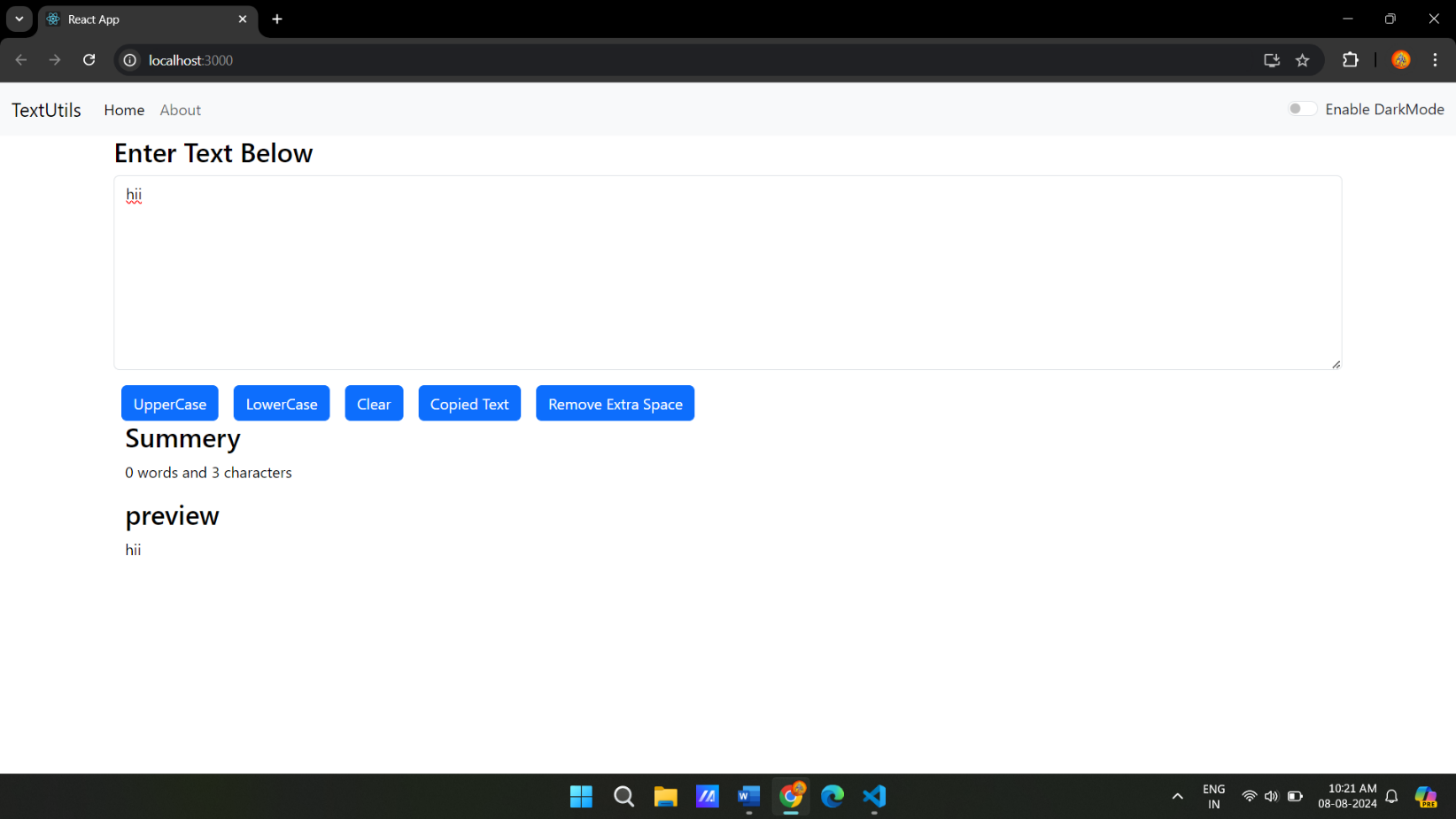
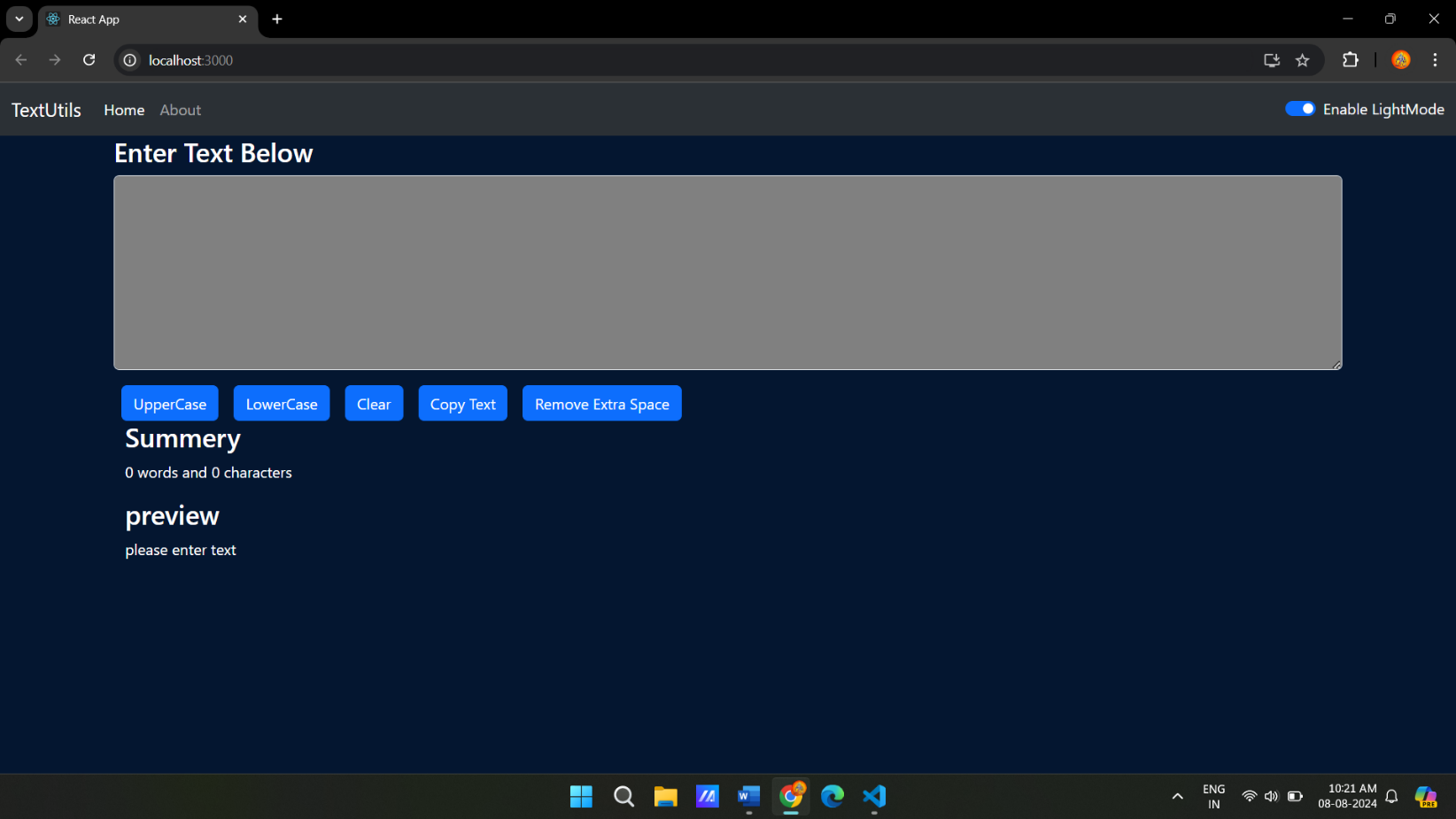
{

    title: PropTypes.string.isRequired,//pts

    home: PropTypes.string.isRequired,

    about: PropTypes.string.isRequired,

}



1. How to show Dismissing alert using state
2. Component to show alert

Alert.js

import React from 'react'

export default function Alert(props) {

    return (

        //both true then execute

        props.alert && <div className={`alert alert-${props.alert.type} alert-dismissible fade show`} role="alert">

            <strong>{props.alert.msg}</strong>

            <button type="button" className="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

        </div>

    )

}

1. Now create state and call it

//app.js

import './App.css';

import About from './components/About';

import Alert from './components/Alert';

import Navbar from './components/Navbar';

import TextForm from './components/TextForm';

import React, { useState } from 'react'//imrs

function App() {

  //set light initialy

  const [theme, setTheme] = useState('light');

  //fun to toogle theme

  function mode() {

    if (theme === 'light') {

      setTheme('dark');

      //universal setting onw set for all

      document.body.style.backgroundColor = '#041630'

      document.body.style.color = 'white'

      //alert

      showAlert("Dark mode is enabled","success");

    }

    else {

      setTheme('light');

      //universal setting

      document.body.style.backgroundColor = 'white'

      document.body.style.color = 'black'

      //alert

      showAlert("Light mode is enabled","success");

    }

  }

  //alert initialy null

  const [alert, setalert] = useState(null);

  //function to show alert

  const showAlert = (message, type) => {

    setalert({

      msg: message,

      type: type,

    })

  }

  return (

    <>

      <Navbar title='TextUtils' home="Home" about="About" theme={theme} mode={mode} />

      <Alert alert={alert} />

      <TextForm theme={theme} />

      {/\* <About /> \*/}

    </>

  );

}

export default App;

1. Now how to make alert is going to auto remove in 1.5sec

Alert.js

import React from 'react'

export default function Alert(props) {

    //fun to uppercase first char

    function upper(word){

        //at 0 upper and join word from 1 to end

        return word.charAt(0).toUpperCase() + word.slice(1);

    }

    return (

        //both true then execute

        props.alert && <div className={`alert alert-${props.alert.type} alert-dismissible fade show`} role="alert">

            <strong>{upper(props.alert.type)} : {props.alert.msg}</strong>

        </div>

    )

}

Textform.js

import React, { useState } from 'react'

export default function TextForm(prop) {

    //in text area always gives us a event

    function handelOnChange(event) {

        setText(event.target.value);

    };

    // Returns a stateful value, and a function to update it.

    const [text, setText] = useState("");

    //upper case

    function onClickUpper() {

        let newText = text.toUpperCase();

        setText(newText);

        prop.showAlert("Text Converted to Uppercase", "success");

    }

    //lower case

    function onClickLower() {

        let newText = text.toLowerCase();

        setText(newText);

        prop.showAlert("Text Converted to Lowercase", "success");

    }

    //clear text

    function onClickClear() {

        setText('');

        prop.showAlert("Text Is cleared", "danger");

    }

    //copy text

    const [copy, setCopy] = useState("Copy");

    function onClickCopy() {

        let newText = document.getElementById('exampleTextarea1');

        // newText.select();

        copy === 'Copy' ? setCopy("Copied") : setCopy("Copy");

        navigator.clipboard.writeText(newText.value);

        prop.showAlert("Text Copied to clipboard", "success");

    }

    //remove extra space

    function onClickRemoveExtraSpace() {

        let newText = text.split(/[  ]+/);

        setText(newText.join(" "));

        prop.showAlert("Extra spaces are removed", "success");

    }

    return (

        <>

            Code

Code

        </>

    )

}

App.js

import './App.css';

import About from './components/About';

import Alert from './components/Alert';

import Navbar from './components/Navbar';

import TextForm from './components/TextForm';

import React, { useState } from 'react'//imrs

function App() {

  //set light initialy

  const [theme, setTheme] = useState('light');

  //fun to toogle theme

  function mode() {

    if (theme === 'light') {

      setTheme('dark');

      //universal setting onw set for all

      document.body.style.backgroundColor = '#041630'

      document.body.style.color = 'white'

      //alert

      showAlert("Dark mode is enabled", "success");

    }

    else {

      setTheme('light');

      //universal setting

      document.body.style.backgroundColor = 'white'

      document.body.style.color = 'black'

      //alert

      showAlert("Light mode is enabled", "success");

    }

  }

  //alert initialy null

  const [alert, setalert] = useState(null);

  //function to show alert

  const showAlert = (message, type) => {

    setalert({

      msg: message,

      type: type,

    });

    //setting alert null in 1.5 sec

    setTimeout(() => {

      setalert(null);

    }, 1500);

  }

  return (

    <>

      <Navbar title='TextUtils' home="Home" about="About" theme={theme} mode={mode} />

      <Alert alert={alert} />

      <TextForm theme={theme} showAlert={showAlert} />

      {/\* <About /> \*/}

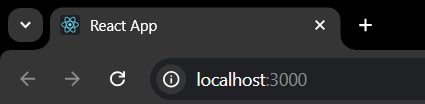
    </>

  );

}

export default App;

1. How to change this title



//my-app1\public\index.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8" />

  <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <meta name="theme-color" content="#000000" />

  <meta name="description" content="Web site created using create-react-app" />

  <link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

  <link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

  <title>TextUtils - Home</title>

//code

//code



1. Now how to change title dynamically

Using document.titile=” ”

App.js

//fun to toogle theme

  function mode() {

    if (theme === 'light') {

      setTheme('dark');

      //universal setting onw set for all

      document.body.style.backgroundColor = '#041630'

      document.body.style.color = 'white'

      //alert

      showAlert("Dark mode is enabled", "success");

      //changing title

      document.title="TextUtils-Dark";

    }

    else {

      setTheme('light');

      //universal setting

      document.body.style.backgroundColor = 'white'

      document.body.style.color = 'black'

      //alert

      showAlert("Light mode is enabled", "success");

      //changing title

      document.title="TextUtils-Light";

    }

  }

How to create an intervals

 //changing titile in itervals

      setInterval(() => {

        document.title = "TextUtils-Amazing";

      }, 1500);

      setInterval(() => {

        document.title = "TextUtils-download";

      }, 2000);

1. Now lets dive into routes

[https://v5.reactrouter.com/web/guides/quick-start](https://v5.reactrouter.com/web/guides/quick-start )

1. Install react router dom(not built-in in react)

npm install react-router-dom

1. Import it in App.js like

import {

  BrowserRouter as Router,

  Switch,

  Route,

  Link

} from "react-router-dom";

1. Now use it

//app.js

import './App.css';

import About from './components/About';

import Alert from './components/Alert';

import Navbar from './components/Navbar';

import TextForm from './components/TextForm';

import React, { useState } from 'react'//imrs

import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';

function App() {

  //code

//code

  return (

    <>

      <Router>

        {/\* content in every page  \*/}

        <Navbar title='TextUtils' home="Home" about="About" theme={theme} mode={mode} />

        <Alert alert={alert} />

        {/\* routes \*/}

        <Routes>

          <Route path='/' element={<TextForm theme={theme} showAlert={showAlert} />} />

          <Route path='/about' element={<About />} />

        </Routes>

      </Router>

    </>

  );

}

export default App;

1. Now link pages(Link not link)

  <Link className="navbar-brand" to="/">{prop.title}</Link>

//navbar.js

import React from 'react'; //imr

import PropTypes from 'prop-types'; //impt

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

//rfc ==> react function based component

export default function Navbar(prop) {

    return (

        <nav className={`navbar navbar-expand-lg bg-body-tertiary bg-${prop.theme === 'light' ? 'dark' : 'light'}`} data-bs-theme={prop.theme}>

            <div className="container-fluid">

                <Link className="navbar-brand" to="/">{prop.title}</Link>

                <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

                    <span className="navbar-toggler-icon"></span>

                </button>

                <div className="collapse navbar-collapse" id="navbarSupportedContent">

                    <ul className="navbar-nav me-auto mb-2 mb-lg-0">

                        <li className="nav-item">

                            <Link className="nav-link active" aria-current="page" to="/">{prop.home}</Link>

                        </li>

                        <li className="nav-item">

                            <Link className="nav-link" to="/about">{prop.about}</Link>

                        </li>

                    </ul>

                    <div className="form-check form-switch">

                        <input className="form-check-input" onClick={prop.mode} type="checkbox" role="switch" id="flexSwitchCheckDefault" />

                        <label className={`form-check-label text-${prop.theme === 'light' ? 'dark' : 'light'}`} htmlFor="flexSwitchCheckDefault" >{prop.theme === 'light' ? 'Enable DarkMode' :'Enable LightMode'}</label>

                    </div>

                </div>

            </div>

        </nav>

    )

}

Navbar.propTypes =

{

    title: PropTypes.string.isRequired,//pts

    home: PropTypes.string.isRequired,

    about: PropTypes.string.isRequired,

}

1. Now how to fix a bug
2. Fixing alert box

Mb-2:adding margin bottom

import React from 'react'

export default function Alert(props) {

    //fun to uppercase first char

    function upper(word) {

        //at 0 upper and join word from 1 to end

        return word.charAt(0).toUpperCase() + word.slice(1);

    }

    return (

        <div className='mb-2' style={{ height: '50px' }}>

            {/\* //both true then execute \*/}

            {props.alert && <div className={`alert alert-${props.alert.type} alert-dismissible fade show`} role="alert">

                <strong>{upper(props.alert.type)} : {props.alert.msg}</strong>

            </div>}

        </div>

    )

}

1. Disabling button when there is no text in textfeild

         <button disabled={text.length === 0} type="button" className="btn btn-primary mx-2 my-2" onClick={onClickLower}>LowerCase</button>

                    <button disabled={text.length === 0} type="button" className="btn btn-primary mx-2 my-2" onClick={onClickClear}>Clear</button>

                    <button disabled={text.length === 0} type="button" className="btn btn-primary mx-2 my-2" onClick={onClickCopy}>{copy} Text</button>

                    <button disabled={text.length === 0} type="button" className="btn btn-primary mx-2 my-2" onClick={onClickRemoveExtraSpace}>Remove Extra Space</button>

                    <button disabled={text.length === 0} type="button" className="btn btn-primary mx-2 my-2" onClick={onClickUpper}>UpperCase</button>

1. Now checking text utils

 <h3>Summery</h3>

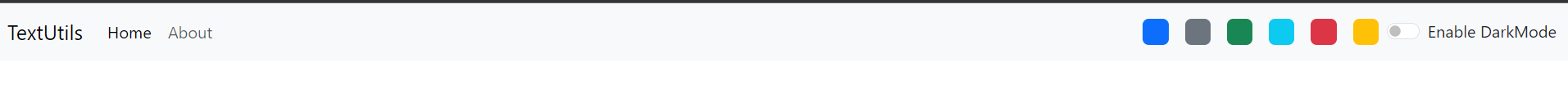
{/\* filtering means returning array of element which contain least one char  \*/}

<p>{text.split(/\s+/).filter((eachElement) => { return eachElement.length !== 0 }).length} words and {text.trim().length} characters</p>

<h3>preview</h3>

<p>{text.length === 0 ? "Nothing To Preview!" : text}</p>

1. Exercise 2 changing colour with js



1. Create a div with d-flex class
2. Insert colors and onclick funcs

 <div className="d-flex" >

                        <div className="bg-primary rounded mx-2" onClick={() => { prop.mode('primary') }} style={{ height: '25px', width: '25px', cursor: 'pointer' }}></div>

                        <div className="bg-secondary rounded mx-2" onClick={() => { prop.mode('secondary') }} style={{ height: '25px', width: '25px', cursor: 'pointer' }}></div>

                        <div className="bg-success rounded mx-2" onClick={() => { prop.mode('success') }} style={{ height: '25px', width: '25px', cursor: 'pointer' }}></div>

                        <div className="bg-info rounded mx-2" onClick={() => { prop.mode('info') }} style={{ height: '25px', width: '25px', cursor: 'pointer' }}></div>

                        <div className="bg-danger rounded mx-2" onClick={() => { prop.mode('danger') }} style={{ height: '25px', width: '25px', cursor: 'pointer' }}></div>

                        <div className="bg-warning rounded mx-2" onClick={() => { prop.mode('warning') }} style={{ height: '25px', width: '25px', cursor: 'pointer' }}></div>

                    </div>

1. Now change mode fun in app.js

  function removeBgClass() {

    document.body.classList.remove('bg-primary');

    document.body.classList.remove('bg-secondary');

    document.body.classList.remove('bg-success');

    document.body.classList.remove('bg-info');

    document.body.classList.remove('bg-danger');

    document.body.classList.remove('bg-warning');

  }

  function mode(cls) {

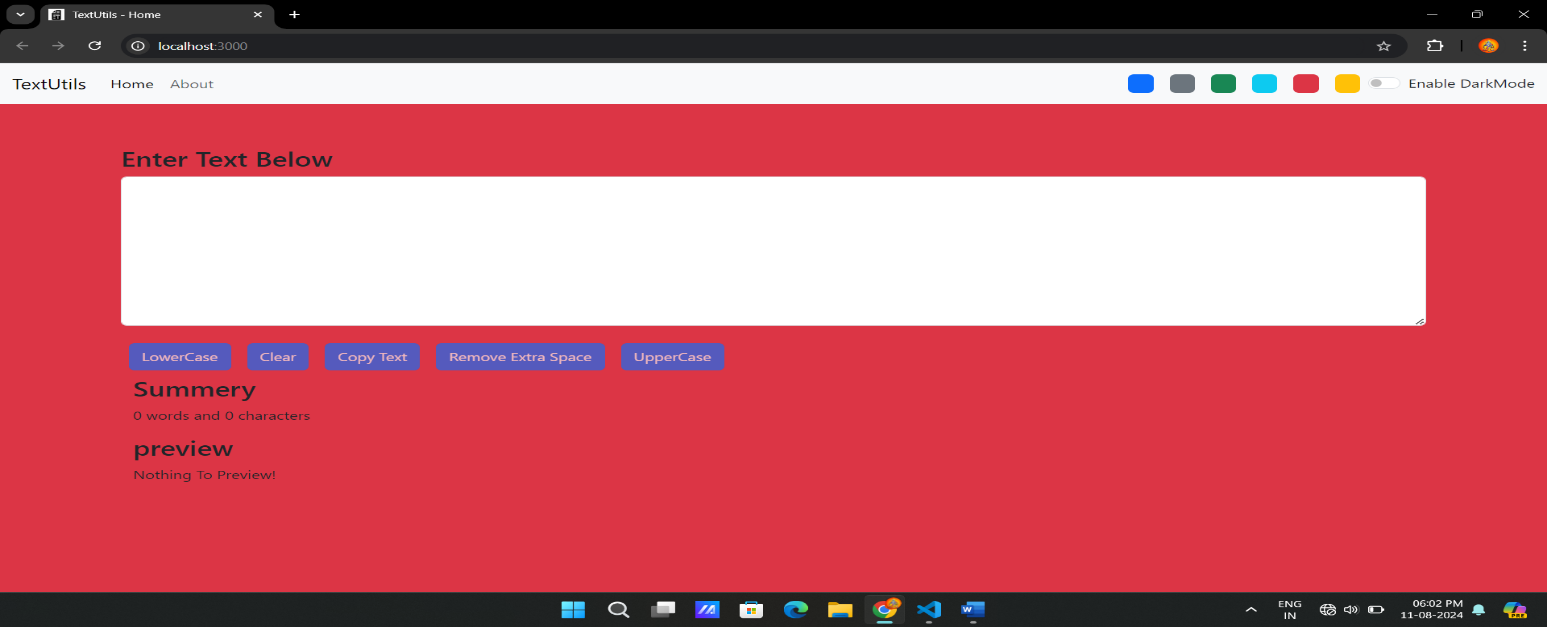
    removeBgClass();

    let temp = `bg-${cls}`;

    document.body.classList.add(temp);

  }

Always need to remove previously added class in body because it uses first in class



1. Now lets go and build monkey app wait but using a react class based component
2. Add bootstrap to index.html
3. And to get snippet of a react class based component use rcc+enter

import './App.css';

//rcc

import React, { Component } from 'react'

export default class App extends Component {

  render() {

    return (

      <div>Hello</div>

    )

  }

}

1. Render in React JS is a fundamental part of class components. It is used to display the component on the UI returned as HTML or JSX components. The ReactDOM.render() function takes two arguments, HTML code and an HTML element.

* Purpose of render()
* React renders HTML to the web page by using a function called render().
* The purpose of the function is to display the specified HTML code inside the specified HTML element.
* In the render() method, we can read props and state and return our JSX code to the root component of our app.
* In the render() method, we cannot change the state, and we cannot cause side effects( such as making an HTTP request to the webserver).

1. Now let’s build navbar

import React, { Component } from 'react'

export default class Navbar extends Component {

    render() {

        return (

            <>

                <nav className="navbar navbar-expand-lg bg-body-tertiary">

                    <div className="container-fluid">

                        <a className="navbar-brand" href="/">MonkeyNews</a>

                        <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

                            <span className="navbar-toggler-icon"></span>

                        </button>

                        <div className="collapse navbar-collapse" id="navbarNav">

                            <ul className="navbar-nav">

                                <li className="nav-item">

                                    <a className="nav-link active" aria-current="page" href="/">Home</a>

                                </li>

                                <li className="nav-item">

                                    <a className="nav-link" href="/">About</a>

                                </li>

                            </ul>

                        </div>

                    </div>

                </nav>

            </>

        )

    }

}

Always change

* href=”#” 🡺 href=”/..”
* <input> 🡺 <input/>
* for 🡺 htmlFor
* class 🡺 className

for using style in jsx

 <div className="container" style={{padding:'20 px', color:"black"}}>

</div>

1. Now lets fetch all data from news api and display it using map
2. First create single news item

//NewsItem.js

import React, { Component } from 'react'

export default class NewsItem extends Component {

    render() {

        //receving propes

        let { title, discription, imageUrl,url } = this.props;

        return (

            <>

                <div className="card" style={{ width: "18rem" }}>

                    {/\* if image is empty then show this image \*/}

                    <img src={!imageUrl?"https://cdn.origo.hu/2024/08/uNiBI1uD0okZN709LDFzNDBJW4rnxNsGjrFP0xdLfdM/fill/1200/675/no/1/aHR0cHM6Ly9jbXNjZG4uYXBwLmNvbnRlbnQucHJpdmF0ZS9jb250ZW50L2ExYzA3NDc5NmRlYzQ2NGViZDE0YWJlZDc4YTg3ZGYy.webp":imageUrl} className="card-img-top" alt="..." />

                    <div className="card-body">

                        <h5 className="card-title">{title}</h5>

                        <p className="card-text">{discription}</p>

                        <a href={url} target='blank' className="btn btn-sm btn-primary">Read More...</a>

                    </div>

                </div>

            </>

        )

    }

}

1. Now using fetch, state, componentDidMount, constructor build a page that display how much total entry are there

//news.js

import React, { Component } from 'react'

import NewsItem from './NewsItem'

export default class News extends Component {

    constructor() {

        super();

        this.state = {

            articles: []

        }

    }

    //Called immediately after a component is mounted. Setting state here will trigger re-rendering.

    async componentDidMount() {

        let url = "https://newsapi.org/v2/everything?q=tesla&from=2024-07-13&sortBy=publishedAt&apiKey=1feeb3ba971c425182492eef91ff97ce";

        let data = await fetch(url);

        let rawData = await data.json();

        this.setState({ articles: rawData.articles });

    }

    render() {

        return (

            <>

                <div className="container my-3">

                    <h3 className=''>

                        MonkeyNews - Top headlines

                    </h3>

                    <div className="row my-3">

                        {this.state.articles.map((e) =>

                            //Warning: Each child in a list should have a unique "key" prop.

                            <div className="col-md-4" key={e.content}>

                                {/\* if any null then show empty string \*/}

                                <NewsItem title={e.title ? e.title : ""} discription={e.description ? e.description : ""} imageUrl={e.urlToImage} url={e.url} />

                            </div>

                        )}

                    </div>

                </div>

            </>

        )

    }

}

1. Now use It in app.js

import './App.css';

//rcc

import React, { Component } from 'react'

import Navbar from './components/Navbar';

import News from './components/News';

export default class App extends Component {

  render() {

    return (

      <div>

        <Navbar/>

        <News/>

      </div>

    )

  }

}