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

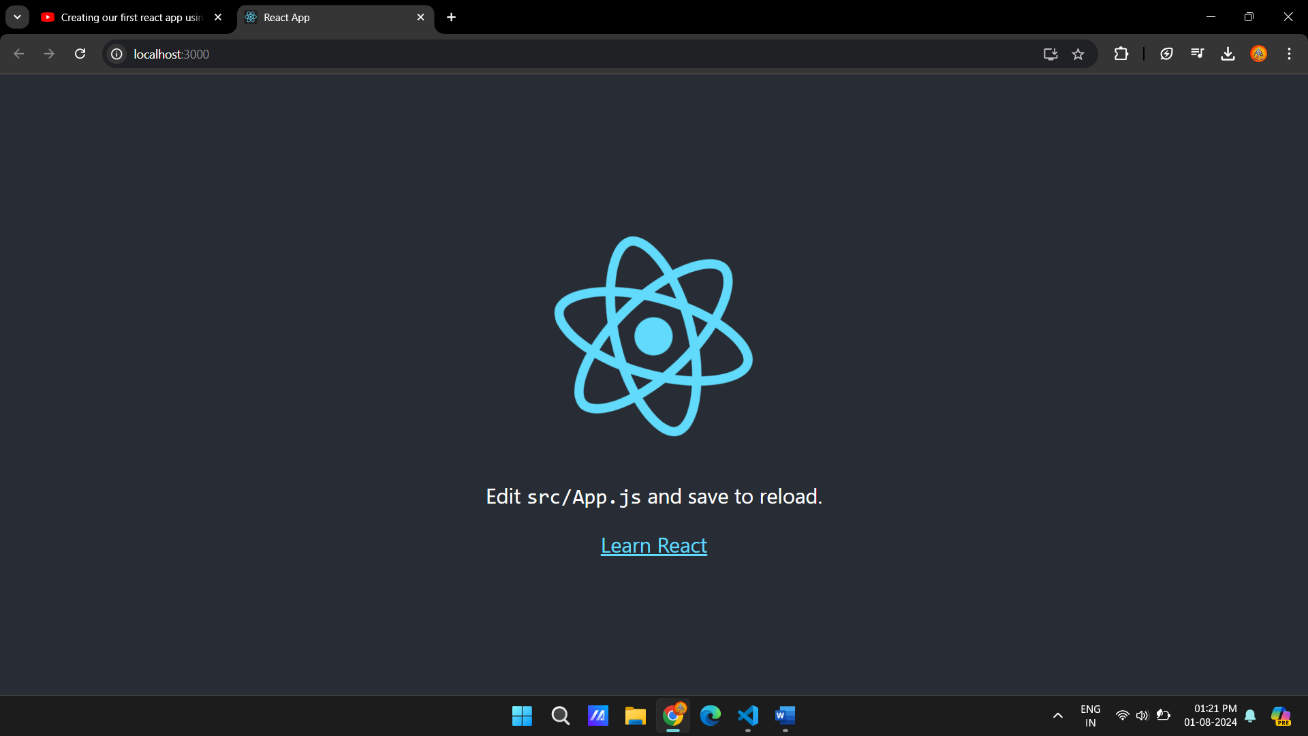
Asus@ASUS-VIVOBOOK-PRO-15 MINGW64 /d/IMPMyWorkplace/React (main)

**$ cd my-app1**

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

**$ npm start**

1. output

****

1. 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>

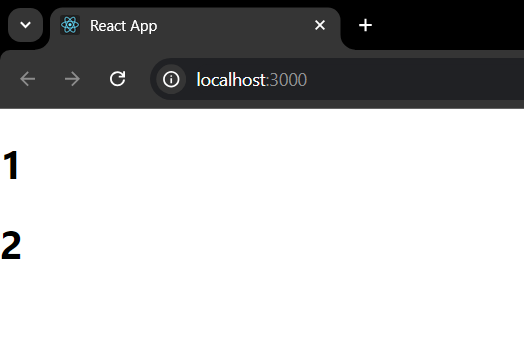
    </>

  );

}

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.

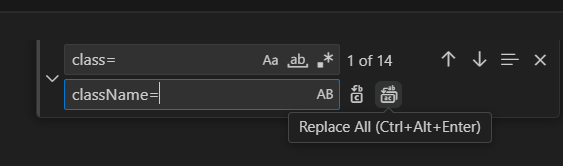
* One more thing is important like we need to change href=” #” to href=”/” because we need to make routes not simply change page or else we get an error

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>

    </>

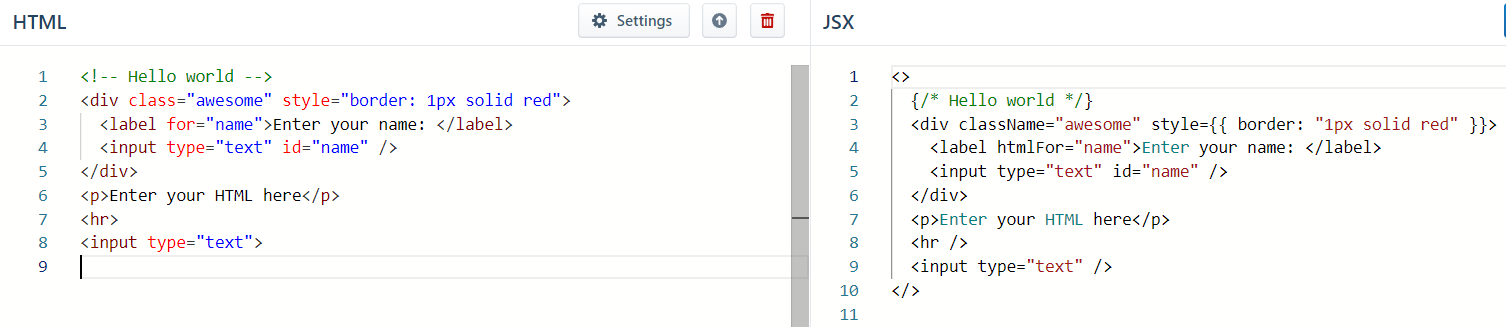
  );

}

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**