**[ GAMECHANGER TIPS AND TRICK 🚀] REACT JS**

**JavaScript Topic use in React.js**

* **Array**
* **Objects**
* **Arrow Function**
* **Map**
* **Filter**
* **Reduce**
* **Mutability and Immutability**
* **Destructuring**

**React.js Topics**

* **Component:** A **component** is a reusable piece of code in React used to build the UI, like a **Navbar**, **Sidebar**, or **Card**. It helps organize the UI into independent, manageable pieces.  
  Components can store their own data using **useState**.
* **Map:** The **map()** function is used to **render lists of components dynamically**. For example, if you want to display 10 cards, you can map over an array of data and render a card for each item.
* **Data Usage - JSX – Dataflow:**
* **JSX (JavaScript XML)** looks like HTML but allows JavaScript expressions using {}

Example: <p>{2 + 2}</p> // Output: 4

* **Dataflow in React is one-way**: data flows **from parent to child** through **props**.
* **Props: Props (Properties)** are used to **pass data or functions from one component to another**, usually from **parent to child**.

**Example use case:**

* Component A sends data to Component B via props.
* B can pass that data to its child C.

If too many levels are involved, this becomes hard to manage. To solve this, we use **Redux** or **Context API**.

* **State - useState: State** is the data **local to a component**. You use **useState** to create and update state. When state changes, the component **automatically re-renders** with the new data.

**Example:**

const [A, setA] = useState(10)

* A represents the current value of the state.
* setA is the function used to update the value of A.
* useState(10) initializes the state with a default value of 10.
* **Effect – useEffect:** useEffectruns after the component mounts (i.e., after it appears on the screen).  
  It’s commonly used for:
* Fetching data from an API
* Running code only once when the page loads
* Updating something when a state or prop changes
* **useRef:** useRef is a React hook that creates a mutable reference to a DOM element or a value. It doesn’t cause re-renders when the value changes. Commonly used to access or manipulate DOM nodes directly.
* **useContext:** React Context is a way to manage state globally. It can be used together with the useState Hook to share state between deeply nested components more easily than with useState alone.
* **Routing:** Routing allows navigation between different views/components. React Router uses <BrowserRouter>, <Route>, and <Link> to manage routes. Helps build single-page applications (SPA) with multiple views.
* **Redux (At the end): Redux** is a state management tool used when **many components need access to the same data**. Instead of passing data down through multiple levels using props, Redux provides a **central store** for global data access. It simplifies the flow of data in large applications.
* **Context API (At the end):** The Context API is a built-in feature in React that allows you to share data between components without passing props manually at every level. It's like a shortcut for props, especially helpful when data needs to be accessed by many components (like themes, language, user info, etc.).
* **Hook:** A **Hook** is a **JavaScript function** that lets you use **React features** (like state and lifecycle methods) **inside functional components**.

|  |  |
| --- | --- |
| **Hook** | **Purpose** |
| **useState** | Adds state to a component |
| **useEffect** | Runs side effects (API calls, timers, etc.) |
| **useContext** | Consumes context values |

* **Common HTML vs React Attribute Differences**

|  |  |
| --- | --- |
| **HTML** | **React (JSX / React Router)** |
| class | className |
| for | htmlFor |
| onclick | onClick |
| onchange | onChange |
| onsubmit | onSubmit |
| onmouseover | onMouseOver |
| onkeydown | onKeyDown |
| tabindex | tabIndex |
| readonly | readOnly |
| maxlength | maxLength |
| autocomplete | autoComplete |
| colspan | colSpan |
| rowspan | rowSpan |
| contenteditable | contentEditable |
| spellcheck | spellCheck |
| autofocus | autoFocus |
| enctype | encType |
| accept-charset | acceptCharset |
| novalidate | noValidate |
| <a href="/about"> | <Link to="/about"> |
| HTML | React (JSX / React Router) |
| class | className |
| for | htmlFor |

* **Fragments:** Fragments let you group multiple elements without adding extra DOM nodes. Instead of wrapping everything in a <div>, you use a React Fragment.

|  |
| --- |
| **Example:**  const Example = () => {  return (  **<>**  <h1>Hello</h1>  <p>This is a fragment</p>  **</>**  );  }; |

* **Axios:** Axios is a JavaScript library used to make HTTP requests (like GET, POST, PUT, DELETE) from the browser or Node.js.

**Complete React JS Course**

**Using ReactJS CDN**

* **React: React** is a **JavaScript library** for building **user interfaces**, mainly for **single-page applications**. It lets you create reusable UI components that update efficiently when your data changes.
* **ReactDOM: ReactDOM** is the package that connects **React** to the **actual DOM** in the browser.

A screenshot of a computer

AI-generated content may be incorrect.

Inside it

A computer screen shot of a computer code

AI-generated content may be incorrect.

Put it into VS Code

A computer screen shot of a black screen

AI-generated content may be incorrect.

Check this through

A black screen with white text

AI-generated content may be incorrect.

Output



**Render H1 using ReactJS CDN**

|  |
| --- |
| **File Path: index.html**  <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Document</title>      <link rel="stylesheet" href="style.css">  </head>  <body>      <div id="parent">      </div>      <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>      <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>      <script src="script.js"></script>  </body>  </html> |

|  |
| --- |
| **File Path: script.js**  var h1 = React.createElement('h1', null, 'Hello ReactJS!');  var parent = document.querySelector('#parent');  var root = ReactDOM.createRoot(parent);  root.render(h1); |

|  |
| --- |
| **File Path: style.css**  \* {      margin: 0;      padding: 0;      box-sizing: border-box;      font-family: gilroy;      color: white;  }  html, body {      height: 100%;      width: 100%;  }  body {      background-color: #111;  } |

**Output:**

A white letter on a black background

AI-generated content may be incorrect.

* Download node JS Pre-build installer
* Check the Node version and create React App command

A screen shot of a computer

AI-generated content may be incorrect.

* Install npm (Node Package Manager)



* File structure of React App **(Don’t touch the node\_modules)**

A screenshot of a computer

AI-generated content may be incorrect.

* Remove the code for src\App.jsx or make it empty
* Use the useState (Hook)

|  |
| --- |
| **File path: src\App.jsx**  import React, { useState } from 'react'  const App = () => {      const [num, setNum] = useState(0)      // const numIncrement = () => {      //     setNum(num + 1)      // }      // const numDecrement = () => {      //     setNum(num - 1)      // }      return (          <div>              <h1>Value of a is {num}</h1>              <button type="button" onClick={()=>setNum(num + 1)}>Increment</button>              <button type="button" onClick={() => setNum(num - 1)}>Decrement</button>              {/\* <button type="button" onClick={numIncrement}>Increment</button>              <button type="button" onClick={numDecrement}>Decrement</button> \*/}          </div>      )  }  export default App |

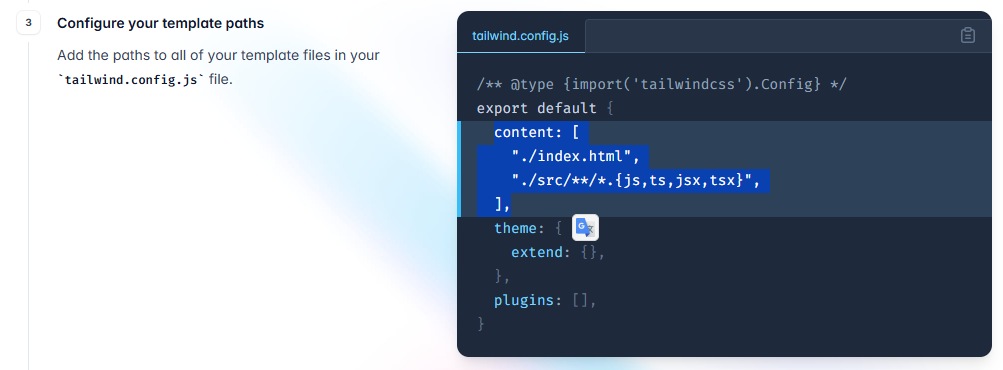
* Add tailwind CSS

A screenshot of a computer

AI-generated content may be incorrect.







|  |
| --- |
| **File Path: tailwind.config.js**  /\*\* @type {import('tailwindcss').Config} \*/  export default {    content: [      "./index.html",      "./src/\*\*/\*.{js,ts,jsx,tsx}",    ],    theme: {      extend: {},    },    plugins: [],  } |

A close up of a computer screen

AI-generated content may be incorrect.

|  |
| --- |
| **File Path: src\index.css**  @tailwind base;  @tailwind components;  @tailwind utilities; |

* If Tailwind CSS suggest are not show, then check

|  |
| --- |
| **File Path: tailwind.config.js**  /\*\* @type {import('tailwindcss').Config} \*/  export default {    content: [      "./index.html",      "./src/\*\*/\*.{js,ts,jsx,tsx}",    ],    theme: {      extend: {},    },    plugins: [],  } |

|  |
| --- |
| **File Path: .vscode\settings.json**  1) .vscode **(File present in project, If not create it)**  2) Create settings.json and paste this code **(Save the file, Close and reopen VS Code)**  {      "tailwindCSS.includeLanguages": {        "javascript": "javascript",        "javascriptreact": "javascript"      },      "editor.quickSuggestions": {        "strings": true      },      "css.validate": false    }    **Now it works (Tailwind suggestions pop up)** |

* **Import CSS file (Create new CSS file):**

Create new CSS file like style.css -> import in **main.jsx** (if you import in main.jsx you can use it anywhere like App.jsx)

Create new CSS file like style.css -> import in **App.jsx** (if you import in App.jsx you can only use it in App.jsx)

A screen shot of a computer code

AI-generated content may be incorrect.

* **Form handling (Stop reload)**

|  |
| --- |
| import React from 'react'  const App = () => {  const submitHandler = (**e**) => {  **e.preventDefault();**  console.log("Hello");  }  return (  <div>  <form onSubmit={(**e**)=>{  submitHandler(**e**)  }}>  <input type="text" className="m-9 px-4 py-2 border border-gray-300 rounded-lg" placeholder="Enter your name" />  <button type="submit" className="m-9 px-4 py-2 bg-blue-600 text-white rounded-lg">Submit</button>  </form>  </div>  )  }  export default App |

* **Two-way binding (**we direct not give input, instead we use **useState)**

|  |
| --- |
| **File Path: src\App.jsx**  import React, { useState } from 'react'  const App = () => {    **const [username, setUsername] = useState('')**      return (      <div>          <form>              <input  **value={username}**  **onChange={(e)=>{**  **setUsername(e.target.value)**  **}}**              className="m-9 px-4 py-2 border border-gray-300 rounded-lg" type="text"              placeholder="Enter your name" />              <button type="submit" className="m-9 px-4 py-2 bg-blue-600 text-white rounded-lg">Submit</button>          </form>      </div>    )  }  export default App |

**Form submit testing using onSubmit Function**

|  |
| --- |
| **File Path: src\App.jsx**  import React, { useState } from 'react'  const App = () => {    **const [username, setUsername] = useState('')**    **const submitHandler = (e)=>{**  **e.preventDefault() // Prevent the default form submission and page reload behavior**  **console.log("Submitted", username)**  **setUsername('') // Clear the input field after submission**  **}**    return (      <div>          <form **onSubmit={(e)=>{**  **submitHandler(e)**  **}}**>  <input  **value={username} // Bind the input value to the username state**  **onChange={(e) => { setUsername(e.target.value) }} // Update the username state with the input value**            type="text"            placeholder="Enter your name"            className="m-9 px-4 py-2 border border-gray-300 rounded-lg"          />              <button type="submit" className="m-9 px-4 py-2 bg-blue-600 text-white rounded-lg">Submit</button>            </form>      </div>    )  }  export default App |

* **Fragments**

|  |
| --- |
| import React, { useState } from 'react'  const App = () => {  return (  // React Fragment is used to group multiple elements without adding extra nodes to the DOM  <>  <h1>Hello</h1>  <h2>Fragements</h2>  </>  )  }  export default App |

* **Components:** Create folder inside src **(src\components)** use small letter

**1)** File inside the components folder, start with capital letter like **“src\components\Header.jsx”**

|  |
| --- |
| **File Path: src\components\Navbar.jsx**  import React from 'react'  const Navbar = () => {    return (      <>        <nav className='flex items-center justify-between bg-gray-800 p-4 text-2xl'>          <h2>Amandeep</h2>          <div className='flex space-x-4 gap-8 items-center'>            <h4>About</h4>            <h4>Contact</h4>            <h4>Services</h4>            <h4>Your Account</h4>          </div>        </nav>      </>    )  }  export default Navbar |

|  |
| --- |
| **File Path: src\App.jsx**  import React from 'react'  **import Navbar from './components/Navbar'**  **import Footer from './components/Footer'**  const App = () => {    return (  **<>**  **<Navbar />**  **<Footer />**  **</>**    )  }  export default App |

* **Props**

|  |
| --- |
| **File Path: src\App.jsx**  import React from 'react'  import Card from './components/Card'  const App = () => {  **const user = 'Amandeep';**    return (      <>  **<Card user={user} /> // Passing the user prop to Card component**      </>    )  }  export default App |

|  |
| --- |
| **File Path: src\components\Card.jsx**  import React from 'react'  **const Card = (props) => { // Destructuring props to get user**    return (  **<div className='text-5xl'>Username is {props.user}</div> // Using the user prop to display the username**    )  }  export default Card |

* Use Json Data through **(forEach() loop and Map())**

|  |
| --- |
| **File Path: src\App.jsx**  import React from 'react'  import Card from './components/Card'  const App = () => {  **const user = [**  **{**  **"name": "Aman Sharma",**  **"city": "Delhi",**  **"age": 28,**  **"profession": "Frontend Developer",**  **"profilePhoto": "https://randomuser.me/api/portraits/men/11.jpg"**  **},**  **{**  **"name": "Sneha Patel",**  **"city": "Mumbai",**  **"age": 24,**  **"profession": "UI/UX Designer",**  **"profilePhoto": "https://randomuser.me/api/portraits/women/21.jpg"**  **}**  **];**  **// user.forEach((element) => {**  **// console.log(element.name, element.city, element.age, element.profession, element.profilePhoto);**  **// });**  **user.map((element)=>{**  **console.log(element.name, element.city, element.age, element.profession, element.profilePhoto);**  **});**  return (  <>  <div className='p-10'>  <Card />  </div>  </>  )  }  export default App  **Output:** |

**Second Way**

|  |
| --- |
| **File path: src\App.jsx**  import React from 'react'  import Card from './components/Card'  const App = () => {  **const user = [**  **{**  **"name": "Aman Sharma",**  **"city": "Delhi",**  **"age": 28,**  **"profession": "Frontend Developer",**  **"profilePhoto": "https://randomuser.me/api/portraits/men/11.jpg"**  **},**  **{**  **"name": "Sneha Patel",**  **"city": "Mumbai",**  **"age": 24,**  **"profession": "UI/UX Designer",**  **"profilePhoto": "https://randomuser.me/api/portraits/women/21.jpg"**  **}**  **];**    return (      <>        <div className='p-10'>  **{user.map((element)=>{**  **return <h1>{element.name}</h1>**  **})}**        </div>      </>    )  }  export default App  **Output:** |

* **Final way to send Json data through props (App.jsx) -> to components (Card.jsx) -> to their child components (Nav.jsx)**

|  |
| --- |
| **File Path: src\App.jsx**  import React from 'react'  import Card from './components/Card'  const App = () => {  **const user = [ // JSON data for users**  **{**  **"name": "Aman Sharma",**  **"city": "Delhi",**  **"age": 28,**  **"profession": "Frontend Developer",**  **"profilePhoto": "https://randomuser.me/api/portraits/men/11.jpg"**  **},**  **{**  **"name": "Sneha Patel",**  **"city": "Mumbai",**  **"age": 24,**  **"profession": "UI/UX Designer",**  **"profilePhoto": "https://randomuser.me/api/portraits/women/21.jpg"**  **},**  **{**  **"name": "Rahul Mehta",**  **"city": "Bangalore",**  **"age": 30,**  **"profession": "Backend Developer",**  **"profilePhoto": "https://randomuser.me/api/portraits/men/45.jpg"**  **},**  **{**  **"name": "Priya Verma",**  **"city": "Pune",**  **"age": 26,**  **"profession": "Full Stack Developer",**  **"profilePhoto": "https://randomuser.me/api/portraits/women/44.jpg"**  **},**  **{**  **"name": "Karan Singh",**  **"city": "Hyderabad",**  **"age": 32,**  **"profession": "DevOps Engineer",**  **"profilePhoto": "https://randomuser.me/api/portraits/men/33.jpg"**  **}**  **];**    return (      <>        <div className='p-3'>  **{user.map((element, idx)=>{**  **return <Card key={idx} username={element.name} age={element.age} title={element.profession} city={element.city} img={element.profilePhoto}/> // Passing props to Card component**  **})}**        </div>      </>    )  }  export default App  **File Path: src\components\Card.jsx**  import React from 'react'  import Nav from './Nav';  const Card = **(props)** => { // Destructuring props to get user  **console.log(props);**    return (      <div className='m-1 bg-white text-black inline-block rounded-lg p-6 w-64 text-center'>          <img className='ml-10 h-32 w-32 rounded-full mb-3' **src={props.img}** alt="" srcSet="" />          <h1 className='text-2xl font-semibold mb-4'>**{props.username}**</h1>          <h1 className='text-xl font-semibold mb-4'>**{props.title}**</h1>          <h2>**{props.city}, {props.age}**</h2>          <button type="button" className='mt-5 bg-emerald-700 text-white px-4 py-2 rounded font-medium'>Add Friend</button>          <Nav username=**{props.username}** /> // Passing props to Nav component      </div>    )  }  export default Card  **File Path: src\components\Nav.jsx**  import React from 'react'  const Nav = **(props)** => {    return (      <div>**{props.username}**</div>    )  }  export default Nav |

* **API Calling with Axios**



|  |
| --- |
| **File Path: src\App.jsx**  import axios from 'axios'  import React, { useState } from 'react'  const App = () => {  **const [data, setData] = useState([])**  **const getData = async () => {**  **const response = await axios.get('https://picsum.photos/v2/list?page=2&limit=10')**  **setData(response.data)**  **console.log(data)**  **}**    return (  **<div className='m-4'>**  **<button onClick={getData} type="button" className='bg-slate-500 text-white p-2 m-2 rounded-xl active:scale-95'>Get Data</button>**  **<div>**  **{data.map((data, idx) => {**  **return (**  **<div key={idx} className='m-1 bg-white text-black inline-block rounded-lg p-6 w-64 text-center'>**  **<img className='ml-10 h-32 w-32 mb-3' src={data.download\_url} alt="" srcSet="" />**  **<h1 className='text-2xl font-semibold mb-4'>{data.author}</h1>**  **<h1 className='text-xl font-semibold mb-4'>{data.title}</h1>**  **</div>**  **)**  **})}**    **</div>**  **</div>**    )  }  export default App |

* **useEffect**

|  |
| --- |
| **File Path: src\App.jsx**  import axios from 'axios'  import React, { useEffect, useState } from 'react'  const App = () => {  const [data, setData] = useState([])    const getData = async () => {      const response = await axios.get('https://picsum.photos/v2/list?page=2&limit=10')      setData(response.data)      console.log(data)    }  **useEffect(() => {**  **getData();**    **}, [])**      return (      <div className='m-4'>        <div>          {data.map((data, idx) => {            return (              <div key={idx} className='m-1 bg-white text-black inline-block rounded-lg p-6 w-64 text-center'>                <img className='ml-10 h-32 w-32 mb-3' src={data.download\_url} alt="" srcSet="" />                <h1 className='text-2xl font-semibold mb-4'>{data.author}</h1>                <h1 className='text-xl font-semibold mb-4'>{data.title}</h1>              </div>            )          })}        </div>      </div>    )  }  export default App |

* **React Router DOM**



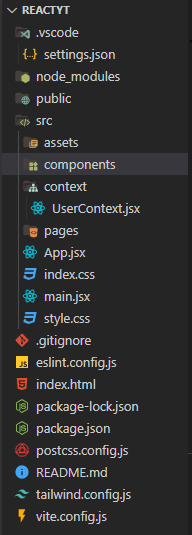
|  |
| --- |
| **File Path: src\main.jsx**  import { StrictMode } from 'react'  import { createRoot } from 'react-dom/client'  import './index.css'  import App from './App.jsx'  **import { BrowserRouter } from 'react-router-dom'**  createRoot(document.getElementById('root')).render(  **<BrowserRouter>**  **<App />**  **</BrowserRouter>**  )  **File Path: src\App.jsx**  import React from 'react'  **import { Route, Routes } from 'react-router-dom'**  **import Home from './pages/Home.jsx';**  **import About from './pages/About.jsx';**  const App = () => {    return (      <div>  **<Routes>**  **<Route path="/" element={<Home />} />**  **<Route path="/about" element={<About />} />**  **</Routes>**      </div>    )  }  export default App |

* **Stop the page reload in routing use <Link to="/"></ Link>**
* **Router + Navbar**

|  |
| --- |
| **File Path: src\main.jsx**  import { StrictMode } from 'react'  import { createRoot } from 'react-dom/client'  import './index.css'  import App from './App.jsx'  **import { BrowserRouter } from 'react-router-dom'**  createRoot(document.getElementById('root')).render(  **<BrowserRouter>**      <App />  **</BrowserRouter>**  )  **File Path: src\App.jsx**  import React from 'react'  **import { Route, Routes } from 'react-router-dom'**  **import Home from './pages/Home.jsx';**  **import About from './pages/About.jsx';**  **import Contact from './pages/Contact.jsx';**  **import Product from './pages/Product.jsx';**  **import Nav from './components/Nav.jsx';**  const App = () => {    return (  **<div>**  **<Nav />**  **<Routes>**  **<Route path="/" element={<Home />} />**  **<Route path="/about" element={<About />} />**  **<Route path="/Contact" element={<Contact />} />**  **<Route path="/Product" element={<Product />} />**  **</Routes>**  **</div>**    )  }  export default App  **File Path: src\components\Nav.jsx**  import React from 'react'  **import { Link } from 'react-router-dom'**  const Nav = () => {    return (  **<div className='py-7 px-10 bg-emerald-600 text-white text-2xl flex items-center justify-between'>**  **<h2>Amandeep <input type="text" className='text-black' /></h2>**  **<div>**  **<ul className='flex gap-10'>**  **<Link to="/" className='cursor-pointer hover:text-emerald-900'>Home</Link>**  **<Link to="/about" className='cursor-pointer hover:text-emerald-900'>About</Link>**  **<Link to="/contact" className='cursor-pointer hover:text-emerald-900'>Contact</Link>**  **<Link to="/product" className='cursor-pointer hover:text-emerald-900'>Product</Link>**  **</ul>**  **</div>**  **</div>**    )  }  export default Nav  **File Path: src\pages\Home.jsx**  import React from 'react'  const Home = () => {    return (      <div>Home</div>    )  }  export default Home  **File Path: src\pages\Contact.jsx**  import React from 'react'  const Contact = () => {    return (      <div>Contact</div>    )  }  export default Contact  **File Path: src\pages\About.jsx**  import React from 'react'  const about = () => {    return (      <div>about</div>    )  }  export default about  **File Path: src\pages\Product.jsx**  import React from 'react'  const Product = () => {    return (      <div>Product</div>    )  }  export default Product  **Create Router in App.jsx -> Give Router link in Nav.jsx -> create page Home.jsx, About.jsx, Product.jsx, Contact.jsx** |

* **Context API**

(context Folder)



|  |
| --- |
| **File Path: src\context\UserContext.jsx**  import React, { createContext } from 'react'  **export const DataContext = createContext()**  **const UserContext = (props) => {**  **const userData = {**  **name: 'John Doe',**  **email: 'John@gamil.com',**  **age: 30,**  **city: 'New York',**  **}**  **return (**  **<div>**  **<DataContext.Provider value={userData}>**  **{props.children}**  **</DataContext.Provider>**  **</div>**  **)**  **}**  export default UserContext  **File Path: src\main.jsx**  import { StrictMode } from 'react'  import { createRoot } from 'react-dom/client'  import './index.css'  import App from './App.jsx'  import { BrowserRouter } from 'react-router-dom'  import UserContext from './context/UserContext.jsx'  createRoot(document.getElementById('root')).render(  **<UserContext>**  **<BrowserRouter>**  **<App />**  **</BrowserRouter>**  **</UserContext>**  )  **File Path: src\App.jsx**  **import React, { useContext } from 'react'**  import { Route, Routes } from 'react-router-dom'  import Home from './pages/Home.jsx';  import About from './pages/About.jsx';  import Contact from './pages/Contact.jsx';  import Product from './pages/Product.jsx';  import Nav from './components/Nav.jsx';  **import { DataContext } from './context/UserContext.jsx';**  const App = () => {  **const data = useContext(DataContext)**  **console.log(data)**    return (      <div>        <Nav />        <Routes>          <Route path="/" element={<Home />} />          <Route path="/about" element={<About />} />          <Route path="/Contact" element={<Contact />} />          <Route path="/Product" element={<Product />} />        </Routes>  **<h3>This will display the username age '{data.age}' from UserContext (App.jsx)</h3>**      </div>    )  }  export default App  **File Path: src\pages\Home.jsx**  **import React, { useContext } from 'react'**  **import { DataContext } from '../context/UserContext'**  const Home = () => {  **const username = useContext(DataContext)**    return (      <div>        <h1>Home</h1>  **<h2>This will display the username '{username.name}' from UserContext</h2>**      </div>    )  }  export default Home  **File Path: src\pages\About.jsx**  **import React, { useContext } from 'react'**  **import { DataContext } from '../context/UserContext'**  const about = () => {  **const username = useContext(DataContext)**    return (      <div>        <h1>About</h1>  **<h2>This will display the username '{username.name}' from UserContext</h2>**      </div>    )  }  export default about |