First) Important VS code Extentions for React ?

\* Auto Rename Tag  
\* Bracket Pair Color DLW

\* ES7 React/Redux/GraphQL/React-Native snippets

\* Git History

\* GitHub Theme

\* GitLens — Git supercharged

\* Live Server  
\* Thunder Client

1) Create React App in a new folder ?

You can Face problem for to Create New App then what will you do ?

1) npm install -g create-react-app

If this process is not working then

2) npm cache clean --force and restart vscode

after doing these

You will did this method

npx create-react-app myapp > Press Enter.

In this case myapp is my app name.

For to create react app without creating new folder

npx create-react-app .

These all process will happen in VS code Terminal or search cmd in your file url section > press enter > put your code

2) Start React app ?

npm start

If node modules is not present in your project then

npm i after doing this npm start

3) What is React Component ?

A React component is a modular and reusable piece of code in the React library that encapsulates specific functionality and can interact with user interfaces through rendering and event handling.

In React, there are two main types of components:

1. **Functional Components:** These are simpler, stateless components that are essentially JavaScript functions. They receive props as input and return React elements.
2. **Class Components:** These are more feature-rich, stateful components that can hold and manage local state. They extend the React.Component class and can include lifecycle methods.

Additionally, with the introduction of React Hooks, functional components can now also handle state and side effects, blurring the distinction between functional and class components.

4) How to create parent component ?

Go to src > App.js >

import React from 'react';

import './App.css';

function App() {

  return (

    <>

      <h1>Welcome to app section</h1>

    </>

  );

}

export default App;

5) Build Parent Child Components :

First create Parent and child components file in your src in a specific folder. Name will be parent.js / parent.jsx

and child.js / child.jsx

i) Parent

import React from 'react';

import Child1 from './Child1';

import Child2 from './Child2';

import Child3 from './Child3';

function Parent() {

  return (

    <>

      <h1>I am your parent</h1>

      <p>Our childs are....</p>

      {/\*Import childs \*/}

      <Child1/>

      <Child2/>

      <Child3/>

    </>

  );

}

export default Parent;

ii) childs:

c1:

import React from 'react';

function Child1() {

  return (

    <>

      <h1>I am Child1</h1>

    </>

  );

}

export default Child1;

c2:

import React from 'react';

function Child2() {

  return (

    <>

      <h1>I am Child2</h1>

    </>

  );

}

export default Child2;

c3:

import React from 'react';

function Child3() {

  return (

    <>

      <h1>I am Child3</h1>

    </>

  );

}

export default Child3;

iii)Grand father / app.js

import React from 'react';

import './App.css';

import Parent from './Component/Parent';

function App() {

  return (

    <>

      <h1>I am your Grand Father</h1>

      {/\*import parent\*/}

      <Parent/>

    </>

  );

}

export default App;

6)Which type of CSS are in React ?

There are three types of CSS are in React :

i)Internal CSS ii) External CSS iii) Inline CSS

i) Internal CSS:

import React from 'react';

//import './App.css';

const App = () =>{

  const headingcolor ={

    color:'red',

    backgroundColor: 'black',

    textAlign: 'center',

    fontSize:'50px',

  }

  return(

    <>

    <h1 style={headingcolor}>Welcome to apps section</h1>

    </>

  )

}

export default App;

ii)External CSS :

a) JSX file : App.js

import React from 'react';

import './App.css';

const App = () =>{

  return(

    <>

    <h1 className='headingdesign'>Welcome to apps section</h1>

    </>

  )

}

export default App;

b) CSS file: App.css

.headingdesign{

  color: yellow;

  background-color: black;

  text-align: center;

  font-weight: bold;

  font-size: 50px;

}

iii) Inline CSS :

import React from "react";

const App = () => {

  return (

    <h1 style={{ color: 'red', backgroundColor: 'black',textAlign:'center'}}>

      I am Inline CSS

    </h1>

  );

}

export default App;

7)How to add image in React ?

Create an image folder in src > enter the image in that particular folder.

import React from 'react';

import exampleImage from './images/example.jpg'; // 'images' is the name of folder where image is stored and 'example.jpg' is the image name.

const App = () => {

  return (

    <>

      <h1>My Name is Mousom Ray</h1>

      <img src={exampleImage} alt="Error" />

    </>

  );

};

export default App;

8)How to add Bootstrap in React ?

After create-react-app go to public > index.html > Put Bootstrap CSS in head section > Put Bootstrap JS in body section.

Lets start project.

9)How to copy paste website template in React ?

**Download a website template document > After extracting the zip file, copy the top folder icon and paste them into your app’s public folder > Open your app’s index.html file in VS Code >Replace the head section of your app’s index.html file with the head section from the template's index.html file > Copy the JS link from the template and paste it at the bottom of your app’s index.html file, inside the body section > Place the body section from the template's index.html file into your app’s components according to component rules.**

10) What is React Router?

React Router is a routing library for React applications, enabling navigation and URL handling within single-page applications (SPAs).

Process of React Rouder:

i) Install React Router:

Go to terminal > npm install react-router-dom

ii) App.js :

import React from 'react'

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

import Home from './Router/Home'

import About from './Router/About'

import Service from './Router/Service'

const App = () => {

  return (

    <>

    <Router>

       {/\* Put Nav always this area \*/}

<Routes>

          <Route path='/' element={<Home />} />

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

          <Route path='/service' element={<Service />} />

        </Routes>

{/\* Put Footer always this area \*/}

      </Router>

    </>

  )

}

export default App

ii) Home.jsx :

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

//import { NavLink } from "react-router-dom"; [For use NavLink]

import React from 'react'

const Home = () => {

  return (

    <div>

      <h1>Home Page</h1>

      <Link to='/about'>About</Link><br />

      <Link to='/service'>Service</Link>

    </div>

  )

}

export default Home

iii) About.jsx :

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

import React from 'react'

const About = () => {

  return (

    <div>

      <h1>About Page</h1>

      <Link to='/'>Home</Link><br />

      <Link to='/service'>Service</Link>

    </div>

  )

}

export default About

iii) Service.jsx :

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

import React from 'react'

const Service = () => {

  return (

    <div>

      <h1>Service Page</h1>

      <Link to='/'>Home</Link><br />

      <Link to='/about'>About</Link>

    </div>

  )

}

export default Service

Note: Always give Nav link path in small letter just like: to='/about' this not to='/About' this

11) Why we use <Link> not <a> ?

We use <Link> instead of anchor tags because <Link> is provided by React Router and ensures client-side routing without full page reloads in a single-page application (SPA) context.

12) What is the difference between <NavLink> and <Link> in react ? and where I use <NavLink> and where I use <Link> ?

React এ `<NavLink>` এবং `<Link>` এর মধ্যে পার্থক্য দুটির মূল বিষয় হ'ল:

1. `<Link>`: `<Link>` হ'ল React Router-এর একটি কোর কম্পোনেন্ট যা ব্যবহার করা হয় পৃষ্ঠাগুলিতে নেভিগেশন করার সময় নতুন URL-এ যেতে। এটি HTML এর `<a>` ট্যাগের মতো কাজ করে, তবে পুরো পৃষ্ঠাটি রিফ্রেশ করেনা এবং কোনো সাধারণ হাইপারলিঙ্কের মতো ব্রাউজার হিসেবে কোনো নতুন রিকোয়েস্ট প্রেরণ করেনা।

2. `<NavLink>`: `<NavLink>` ও একই React Router প্যাকেজের অংশ এবং এটি `<Link>` এর একটি সাব-ক্লাস হিসেবে কাজ করে, তবে এটি একটি একক পৃষ্ঠা সম্পর্কে সম্পূর্ণ সচেতন থাকে এবং সেটি ন্যাভিগেশনের সময় অবশ্যই একটি নির্দিষ্ট ক্লাস অ্যাক্টিভ হয়ে থাকে যেটি বর্তমানে নির্দিষ্ট পৃষ্ঠার নির্দেশক হিসেবে চিহ্নিত করে। এটি CSS অ্যাক্টিভ স্টেট সেট করার সাথে সাথে এটি নির্দিষ্ট স্টাইল ক্লাস অ্যাড করে যা সংশ্লিষ্ট ন্যাভিগেশনের অনুরূপ হতে পারে।

একটি পাতা থেকে অন্য পাতায় যাওয়ার জন্য, আপনি সাধারণত `<Link>` ব্যবহার করবেন। তবে, আপনি যদি একটি নেভিগেশন মেনু তৈরি করতে চান যেখানে বর্তমান পৃষ্ঠাটি কোন নির্দিষ্ট স্থানে থাকবে, তবে আপনি `<NavLink>` ব্যবহার করতে পারেন যাতে সংশ্লিষ্ট লিংক সক্রিয় হওয়ার সময় একটি নির্দিষ্ট স্টাইল অ্যাক্টিভ করা হয়।

**Import process as same as <Link>**

**13) How to create active Nav bar when I will be visit any Nav Area ?**

* I replaced the Link import with NavLink from 'react-router-dom'.
* Each NavLink component now has the activeClassName prop set to "active".
* The exact prop is added to the Home NavLink to ensure it is only active when the exact path matches.

**Home area:**

<NavLink exact to="/" className="nav-item nav-link" activeClassName="active">Home</NavLink>

The exact prop ensures that the NavLink is only considered active when the URL matches the NavLink's path exactly, preventing partial matches from activating it.

Others area:

<NavLink to="/About" className="nav-item nav-link" activeClassName="active">About</NavLink>

**Note:** "When using the **<NavLink>** component, replace all occurrences of **class** with **className**."

**14) What is Props in React ?**

React-এ props হল একটি গুরুত্বপূর্ণ ধারণা যা আমাদের কোডে ডেটা পাস করায় ব্যবহার করা হয়। props একটি React component থেকে অন্য একটি React component-এ ডেটা পাঠানোর মাধ্যমে সংজ্ঞায়িত হলে সেটির সাহায্যে আমরা বিভিন্ন ডাইনামিক UI তৈরি করতে পারি।

props একটি JavaScript object যা প্রথম React component থেকে পরবর্তী React component-এ ডেটা পাঠাতে ব্যবহৃত হয়। এই ডেটা প্রধানত স্ট্রিং, নম্বর, অথবা অন্য কোনো React elements হতে পারে।

যেমন, একটি React component যেখানে আমরা "name" এবং "age" props পাঠাচ্ছি:

```jsx

<MyComponent name="John" age={30} />

```

এখানে "name" এবং "age" হল props এবং এগুলি MyComponent component-এর মধ্যে পাঠানো ডেটা। এই ডেটা MyComponent অবজেক্টে প্রাপ্ত হবে এবং এর ভিতরে ব্যবহৃত হবে।

সুতরাং, props একটি React component-এর সাথে তথ্য পাঠানোর মাধ্যমে আমরা ডাইনামিক UI তৈরি করতে পারি।

**Input string Data in Props**

Basic code of Props:

**String Parent:**

import React from 'react'

import Stringchild from './Stringchild'

const Stringparent = () => {

  return (

    <>

    <h1>I am string Parent</h1>

    <Stringchild name="Mousom" age={22} city="Uttarpara"/>

    </>

  )

}

export default Stringparent

**String Child:**

import React from 'react'

const Stringchild = (props) => {

    //  If you do using destructuring

    const { name, age, city } = props;

    return (

        <>

Using Destructuring

            <h4>I am String Child</h4>

            <h4>Name:{name}</h4>

            <h4>Age:{age}</h4>

            <h4>City:{city}</h4>

Not using destructuring

            {/\*If you not use destructuring\*/}

            <h4>I am String Child</h4>

            <h4>Name:{props.name}</h4>

            <h4>Age:{props.age}</h4>

            <h4>City:{props.city}</h4>

        </>

    )

}

export default Stringchild

**App.js:**

Import Stringparent component in the App.js file as the rules of Import Component

**Input “Array object Data” in Props**

**Arrayparent:**

import React from 'react'

import Arraychild from './Arraychild';

const Arrayparent = () => {

    const mydata = [

        { name: 'Mousom', age: 23, city: 'Uttarpara' },

        { name: 'Ram', age: 30, city: 'Ayodha' },

        { name: 'Samir', age: 62, city: 'Kalna' }

    ];

    return (

        <>

        <h2>I am Array parent</h2>

        {mydata.length>0 ? <Arraychild data={mydata}/> : <p>Data not Found</p>}

        </>

    )

}

export default Arrayparent

This is the code of ternary operator

{mydata.length>0 ? <Arraychild data={mydata}/> : <p>Data not Found</p>}

**Arraychild:**

import React from 'react'

const Arraychild = (props) => {

  return (

    <>

    <h2>I am Arraychild</h2>

    <h4>

    {props?.data?.map((value,index) => {

        return(

        <div key={index}>

        My name is {value.name}. I lived in {value.city} and my age is {value.age}

        </div>

)

    })}

    </h4>

    </>

  )

}

export default Arraychild

This is the syntax of Map :

{props?.data?.map((value,index) => {

return(

<div key={index}>

        My name is {value.name}. I lived in {value.city} and my age is {value.age}

        </div>

)

    })}

**Use Addition and Multiplication using props**

**Parent:**

import React from 'react'

import Numberchild from './Numberchild'

const Numberparent = () => {

  return (

   <>

   <h1>I am Number Parent</h1>

   <Numberchild add={5+4} mul={5\*4}/>

   </>

  )

}

export default Numberparent

**Child:**

import React from 'react'

const Numberchild = (props) => {

  return (

    <>

    <h1>I am Number Child</h1>

    <h2>The addition of 5 and 4 is : {props.add} and Multiplication is : {props.mul}</h2>

    </>

  )

}

export default Numberchild

**Input only object data in props**

**Parent:**

import React from 'react'

import Child from './Child'

const Parent = () => {

  const data = {name:'Mousom',age:23,city:'Uttarpara'};

    return (

    <>

    <Child mydata={data}/>

    </>

  )

}

export default Parent

**Child:**

import React from 'react'

const Child = ({mydata}) => {

  return (

    <>

    <h4>

    Name: {mydata.name }

    Age: {mydata.age }

    City: {mydata.city }

    </h4>

    </>

  )

}

export default Child

**15)What is Params and why we use Params ?**

React Params are used to dynamically pass data between components in React applications, enhancing flexibility and reusability.

**16) How to create React Params ?**

**npm install react-router-dom > create a Nav bar and make router link using the rules of Router.**

**App.js:**

<Router>

        <Nav />

        <Routes>

          <Route path='/feature' element={<Error />} />

          <Route path='/feature/:id/:name' element={<Feature />} />

        </Routes>

      </Router>

**Feature.jsx:**

import React from 'react'

import { useParams } from 'react-router-dom'

const Feature = () => {

    const{id,name} = useParams();

  return (

    <>

    <h3>This is Feature Child section</h3>

    <h5>Id: {id}</h5>

    <h5>Name: {name}</h5>

    </>

  )

}

export default Feature

**Nav.jsx: All are same as previous process.**

**Note:** <Route path='/feature' element={<Error />} />

This code is use for to show 404 Page not Found Message.

<Route path='/feature/:id/:name' element={<Feature />} />

In this code you can change :id or :name to others name or you add another name or other thing.

How to fill id and name are ?

After starting React app > Go to view site information tab > Search <http://localhost:3000/feature/2/Virat> output will be come.

17) What is Hook in React ?

A Hook is a feature in React that allows functional components to use state and other React features without writing a class, offering more flexibility and reusability in component logic.

There are two main types of hooks in React: **State Hooks** and **Effect Hooks**.

18) What is State Hooks ?

State Hooks in React, like useState, enable functional components to manage state locally without using class components.

**Change Name after click button :**

import { useState } from 'react'

const State = () => {

    const defaultName = "Mousom"

    const [myName, setMyName] = useState(defaultName)

    const changeName = () => {

        setMyName("Virat")

    }

    const revertName = () => {

        setMyName(defaultName)

    }

    return (

        <>

            <h1>Use State Hooks {myName}</h1>

            <button className='btn' onClick={changeName}>Click-me</button>

            <button className='btn' onClick={revertName}>Revert</button>

        </>

    )

}

export default State

**Counter :**

import React, { useState } from 'react';

export default function Counter() {

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

  const increment = () => {

    setCount(count + 1);

  };

  const decrement = () => {

    // Check if count is greater than 0 before decrementing

    if (count > 0) {

      setCount(count - 1);

    }

  };

  return (

    <div>

      <h1>Counter: {count}</h1>

      <button onClick={increment}>Increment</button>

      <button onClick={decrement}>Decrement</button>

    </div>

  );

}

**Change color :**

import React from 'react'

import { useState } from 'react'

const Changecolor = () => {

    const[piller,changable] = useState('black'); {/\*useState always stored in first variable  in this case 'piller'\*/}

    const red = () =>{

        changable('red');

    }

    const orange = () =>{

        changable('orange');

    }

    const back = () =>{

        changable('black');

    }

    return (

    <>

    <div style={{backgroundColor:piller,height:'200px'}}>

    </div>

    <button className='btn' onClick={red}>Red</button>

    <button className='btn' onClick={orange}>Orange</button>

    <button className='btn' onClick={back}>Back</button>

    </>

  )

}

export default Changecolor

**19) What is useEffect hook ? And why it is used ?**

The **useEffect** hook in React is used to perform side effects in function components, such as data fetching, subscriptions, or manually changing the DOM, after the component has been rendered.

The **useEffect** hook is used in React to manage side effects in function components, ensuring actions are performed after rendering, like fetching data or interacting with the DOM.

**Alert using useEffect:**

import React from 'react'

import { useState, useEffect } from 'react'

import Practicenav from './Practicenav';

const Effect = () => {

  const [piller, changable] = useState(0);

  const [first, setFirst] = useState(0);

  // Case 1:Run on Every render

  useEffect(() => {

    alert('Hey I will run on every render');

  })

  // Case 2:Run only on First render

  useEffect(() => {

    alert('Hey I will run on First render');

  },[])

  // Case 3:Run only on Certain render

  useEffect(() => {

    alert('I was run because Count was change');

  }, [piller])

  useEffect(() => {

    alert('First was change');

  }, [first])

  const increament = () => {

    changable(piller + 1);

  };

  const decreament = () => {

    if (piller > 0) {

      changable(piller - 1);

    }

  }

  return (

    <>

      <Practicenav color={'navy' + 'blue'} />

      <h1>Counter: {piller}</h1>

      <button className='btn' style={{ backgroundColor: 'blue', color: 'white' }} onClick={increament}>Increament</button>

      <button className='btn' style={{ backgroundColor: 'green', color: 'white' }} onClick={decreament}>Decreament</button>

    </>

  )

}

export default Effect

**Practice Nav code :**

import React, { useEffect } from 'react'

const Practicenav = ({ color }) => {

    useEffect(() => {

        alert('Color was changed')

    }, [color]);

    return (

        <>

            <nav>

                I <h1>I am Navbar of {color} color hehe</h1>

            </nav>

        </>

    )

}

export default Practicenav

**Counter using useEffect:**

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

const Counter = () => {

    const [counttime,setCount]=useState(0)

    useEffect(()=>{

        setTimeout(()=>{

            setCount((count)=>count+1)

        },1000)

    },[]); // Blank Array use to stop counting

return (

    <>

    <h1>Set time out example using useEffect hooks</h1>

    <h1>stats count down {counttime}</h1>

    </>

  )

}

export default Counter

**20) What is [ ] this in react useEffect hook ?**

`[]` এই চিহ্নটি রিয়েক্ট হুকের `useEffect` ফাংশনের দ্বিতীয় প্যারামিটার। এটি একটি ডিপেন্ডেন্সি অ্যারে যা বোঝায় কোন স্টেট বা প্রপার্টির পরিবর্তনের উপর এই `useEffect` এর কাজ নির্ভর করবে।

যদি আপনি `[]` দেখেন, তবে `useEffect` এর কাজ কোন স্টেট বা প্রপার্টির উপর নির্ভর করবে না। অর্থাৎ, এই `useEffect` শুধুমাত্র একবার কল হবে, প্রথম রেন্ডারিং এর পরে। এটি যে কোন প্রকার পরিবর্তনের নির্দিষ্ট না করে এই প্যারামিটার দ্বারা উল্লিখিত কোন ডিপেন্ডেন্সিতে।

এটি একটি কমন প্যাটার্ন যখন আপনি চান না যে কোন ডিপেন্ডেন্সির উপরে নির্ভর করা যেতে পারে, বা যখন আপনি চান না কোন কিছু বদলে `useEffect` এর কাজ হোক না প্রথম রেন্ডারিং পরে।

**21) Why counting is stop after using [ ] this ?**

`[]` ব্যবহার করা `useEffect` হুকের মাধ্যমে কিন্তু সময়ের সাথে সম্পর্কিত কোন নতুন স্টেট বা প্রপার্টির উপর নির্ভর নেই। যখন আপনি `[]` ব্যবহার করেন, তখন সেটাই বুঝায় যে, এই `useEffect` কেবল মাত্র প্রথম রেন্ডারিং পরে একবার কল হবে এবং পরবর্তীতে কোন নতুন রেন্ডারিং এ পরিবর্তন করবে না।

তার মানে হচ্ছে, যখন `count` স্টেট এর মান পরিবর্তিত হবে না, তখন `useEffect` পুনরায় কল হবে না। এই প্রোগ্রামে, `count` এর মান পরিবর্তিত হচ্ছে না, কারণ আমরা `useEffect` এর ডিপেন্ডেন্সির অ্যারে এখানে খালি ছিল। তাই এটি প্রতিবার পরের সেকেন্ডে একটি নতুন `setTimeout` কল করা সাপেক্ষে না।

এই কারণে, এই প্রোগ্রামে সময়ের সাথে সম্পর্কিত গণনা বন্ধ হয়ে যাচ্ছে কারণ `useEffect` পুনরায় কল হচ্ছে না এবং তার কারণে নতুন টাইমার সেট করা হচ্ছে না।

**Fetch API using useEffect:**

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

const ProductFetch = () => {

    const [prod,setProduct]=useState([])

    const getProduct=async()=>{

        const respoinse=await fetch('https://dummyjson.com/products')

        const response=await respoinse.json()

        setProduct(response)

    }

    useEffect(()=>{

        getProduct()

    },[])

    console.log('pp',prod);

  return (

    <>

    <h1>Fetch Product</h1>

    {

        prod?.products?.map((pro,key)=>{

            return (

                <>

               <h1>{key+1}</h1>

                <h1 key ={pro.id}> {pro?.title}</h1>

                </>

            )

        })

    }

    </>

  )

}

export default ProductFetch

**Note:** অনেক সময় map চালালে দেখায় যে prod.map is not a function তো তখন করবো ?

{Array.isArray(prod) && prod?.products?.map

**22) How to create Loading Bar using Use state hook when you fetch data from API ?**

**Step1: Create a state hook for Loading**

**const [loading, setLoading] = useState(true);**

**Step2: setLoading will be false if data is load**

**setData(responseData.products); //Data is load**

**setLoading(false); // So setLoading(false)**

**Step3: Create a Condition before parent function return statement**

**if (loading) {**

**return <h1 style={{ fontSize: '100px', color: 'red', display: 'flex', justifyContent: 'center', alignItems: 'center', height: '100vh' }}>Loading....</h1>;**

**}**

**23) How to Create Load more button for cards?**

**Step1: Create a state hook for Load more button**

**const [visibleCards, setVisibleCards] = useState(6); // 6 means first 6 cards will be visible**

**Step2: Create a function for Load more button before parent function return statement**

**const handleLoadMore = () => {**

**setVisibleCards(prevVisibleCards => prevVisibleCards + 6);**

**};**

**Step3: Add Slicer in map**

**{data?.slice(0, visibleCards).map((product) => {**

**........**

**}**

**Step4: Make Button with condition**

**{visibleCards < data.length ? (**

**<div className="text-center mt-3">**

**<button className="btn btn-primary" onClick={handleLoadMore}>Load More</button>**

**</div>**

**) : null}**

**This condition is to say if All loading are complete then Button will be gone**

**24) When we use dangerouslySetInnerHTML ?**

**dangerouslySetInnerHTML** is a React attribute used to render HTML strings as content in JSX, but its use poses a security risk as it bypasses React's built-in XSS protections.

**Syntax :**

<div dangerouslySetInnerHTML={{

        \_\_html: value.postText

  }} />

In this code value is my map parametre and postText is my properties of API.

**Note: Till Now I use it for to remove html tags from my API value.**

**25) What is Axios in React ?**

Axios is a popular JavaScript library used for making HTTP requests from browser-based applications, including those built with React.

Axios supports HTTP methods such as GET, POST, PUT, DELETE, and more for making requests from client-side JavaScript applications.

How to import Axios ?

Go to VS code terminal > npm i axios

Syntax of Axios Get :

import axios from 'axios';

import React from 'react'

import { useState,useEffect } from 'react'

const AxiosAPI = () => {

const[blog,setblog] = useState([]);

const fetchdata = async() =>{

    const response = await axios.get('https://restapinodejs.onrender.com/api/allBlog');

    setblog(response.data);

}

useEffect(() =>{

    fetchdata();

},[])

  return (

    <>

      {

        blog?.data?.map((value) =>{

            return(

            <div key={value.\_id}>

                <h1>{value?.title}</h1>

                <p

                dangerouslySetInnerHTML={{\_\_html:value?.postText}}

                />

            </div>

        )})

      }

    </>

  )

}

export default AxiosAPI

**Note: We use Axios for to fetch API from server . It is substitute and more easy process than fetch.**

**We will always use ‘**setblog(response.data);’ .data in the time of Axios.

`.data` is a property in Axios that retrieves the response data from an HTTP request.

**26) What is Use Reducer hook ? Why we use it ?**

The **useReducer** hook in React is used for managing complex state logic in applications by centralizing state updates through a reducer function.

**Counter styntax using usereducer( ) :**

import React from 'react'

import { useReducer } from 'react'

const Counter = () => {

    const initialstate = 1;

    const reducer = (state, action) => {

        switch (action.type) {

            case "incriment":

                return state + 3

            case "decriment":

                return state - 1

            case "mult":

                return state \* 2

            default:

                throw new Error()

        }

    }

    const [state, dispach] = useReducer(reducer, initialstate)

    return (

        <>

            <h1>Use Reducer hooks</h1>

            <h1> {state}</h1>

            <button onClick={() => dispach({ type: "incriment" })}>Incriment</button>

            <button onClick={() => dispach({ type: "decriment" })}>DecIncriment</button>

            <button onClick={() => dispach({ type: "mult" })}>Mult</button>

        </>

    )

}

export default Counter

**27) const reducer=(state,action)=>{ what is state and what is action in this code ?**

এখানে reducer হলো একটি ফাংশন যা আমাদের স্টেট পরিচালনা করে। এটি বর্তমান স্টেট এবং কোনো কার্যকলাপের (action) সাথে সম্পর্কিত লজিক নির্ধারণ করে এবং নতুন স্টেট তৈরি করে। রিডাক্স বা অন্য স্টেট ম্যানেজমেন্ট লাইব্রেরিতে, reducer স্টেটের কোনো পরিবর্তন বা কার্যকলাপের অনুমতি বা নিয়ন্ত্রণ প্রদান করে। এটি একটি পুর্যাতন স্টেট এবং অ্যাকশান প্রদান করে, এবং একটি নতুন স্টেট তৈরি করে যা প্রোগ্রামের অবস্থা পরিবর্তন করে।

এই কোডে, `state` হলো বর্তমান অ্যাপ্লিকেশন স্টেট যেটি আমরা পরিচালনা করতে চাই এবং `action` হলো কোনো কার্যকলাপ বা ইভেন্ট যা স্টেটের পরিবর্তন সৃষ্টি করে।

অর্থাৎ, `state` হলো আমাদের অ্যাপ্লিকেশনের বর্তমান অবস্থা এবং `action` হলো একটি ইভেন্ট যা অ্যাপ্লিকেশনের অবস্থা পরিবর্তন করে। রিডাক্স বা অন্য স্টেট ম্যানেজমেন্ট লাইব্রেরি ব্যবহার করার সময়, এই দুটি প্রধান উপাদান হিসেবে ব্যবহার হয়।

**28) What is dispatch ?**

dispatch in useReducer hook is a function used to send actions to the reducer to update the state.

**29) What is throw new Error( ) ?**

`throw new` in React is used to raise an error intentionally within a component or function.

`Error` is a built-in object in JavaScript for representing an error condition.

**30) What is the difference between usestate( ) and usereducer( ) hook ?**

useState হুক এবং useReducer হুক উভয়ই React-এর স্টেট ম্যানেজমেন্টের জন্য ব্যবহৃত হলেও এদের মধ্যে কিছু পার্থক্য রয়েছে।

1. useState হুক:

- useState হুক ব্যবহার করা হয় ছোট ও সাধারণ স্টেট ম্যানেজমেন্টের জন্য।

- এটি একটি সিঙ্গেল স্টেট ম্যানেজ করে।

- স্টেট আপডেট করতে সরাসরি নতুন মান সেট করা হয়।

2. useReducer হুক:

- useReducer হুক ব্যবহার করা হয় বৃহত্তর এবং সংশ্লিষ্ট কমপ্লেক্স লজিকের স্টেট ম্যানেজমেন্টের জন্য।

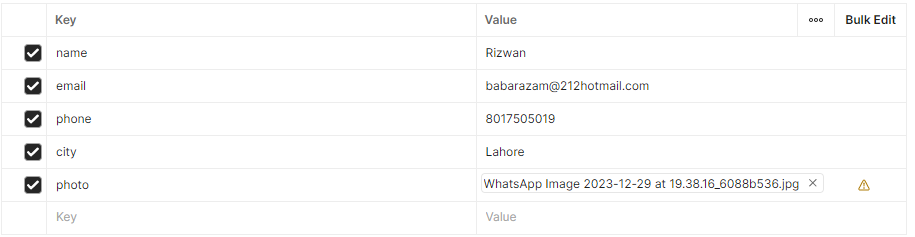
- এটি একটি বিস্তৃত অবস্থানের জন্য ব্যবহৃত হয়, যেখানে স্টেট এবং পরিবর্তনের লজিক একটি reducer ফাংশনের মাধ্যমে পরিচালিত হয়।

- এটি বেশি সংগঠিত ও একটি সমৃদ্ধ ম্যানেজমেন্ট উপায় প্রদান করে যা বৃহত্তর অ্যাপ্লিকেশনগুলিতে দরকারী হতে পারে।

এই উভয় হুক ব্যবহার করে আমরা স্টেট ম্যানেজ করি, তবে সম্প্রতি React ডেভেলপারদের মধ্যে useReducer হুক এর ব্যবহার দেখা পাওয়া যাচ্ছে বৃহত্তর অ্যাপ্লিকেশনগুলির জন্য, যেখানে লজিক প্রয়োজন এবং কনফিগারেশন বেশ কম্পলেক্স।

**31) How to Add Data in Get Api with the help of Post Api using form ?**

First Get a Post API > Put in into Postman Software > Click on left side dropdown ‘v’ > Select Post > Click on Body > First check is it a form data or raw data > Click on form-data > Fill up form like this

****

> If It is a form data output will be like this

If it is not a form data out put will be came like this



> Click on Send blue button.

**32) Put data in Get API using form post api :**

import React from 'react'

import axios from 'axios'

import { useState } from 'react'

//npm i react-toastify [For install React toastify]

import { ToastContainer, toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

const Postformdata = () => {

    // Create initial state for api properties

    const initialstate = {

        name: "",

        email: "",

        city: "",

        phone: "",

        photo: ""

    }

    const [user, setUser] = useState(initialstate)

    const [loading, setLoading] = useState(false); // State for loading indicator

    const [img, setimg] = useState(); // Use State for image

    // Handle for onchange

    const handleChange = (e) => {

        const { name, value } = e.target

        setUser({ ...user, [name]: value })

        console.log(user);

    }

    const handleOnSubmit = async (e) => {

        e.preventDefault()

        // Start loading

        setLoading(true);

        const apiUrl = "https://tureappservar.onrender.com/user";

        // Because it is a form data so I use form data

        let formdata = new FormData()

        formdata.append("name", user.name)

        formdata.append("city", user.city)

        formdata.append("email", user.email)

        formdata.append("phone", user.phone)

        formdata.append("photo", img)

        try {

            const response = await axios.post(apiUrl, formdata)

            if (response && response?.data?.status === 200) {

                console.log(response);

                toast.success(response?.data?.message)

                setLoading(false)

                setUser(initialstate)

            } else {

                toast.error(response?.data?.message)

                setLoading(false)

            }

        } catch (error) {

            console.log(error);

            toast.error(error?.response?.data?.message)

            setLoading(false)

        }

    }

    return (

        <>

            <ToastContainer />

            <h1>Add user</h1>

            <div className="container">

                <form onSubmit={handleOnSubmit} method='post'>

                    <div className="form-group">

                        <label for="exampleInputEmail1">Name</label>

                        {/\* name attribute is mandetory \*/}

                        <input type="text" name='name' className="form-control"

                            onChange={handleChange}

                            value={user.name}

                        />

                    </div>

                    <div class="form-group">

                        <label for="exampleInputEmail1">Email</label>

                        <input type="email" class="form-control" name='email'

                            onChange={handleChange}

                            value={user.email}

                        />

                    </div>

                    <div class="form-group">

                        <label for="exampleInputPassword1">Phone</label>

                        <input type="text" class="form-control" name='phone'

                            onChange={handleChange}

                            value={user.phone}

                        />

                    </div>

                    <div class="form-group">

                        <label for="exampleInputPassword1">City</label>

                        <input type="text" class="form-control" name='city'

                            onChange={handleChange}

                            value={user.city}

                        />

                    </div>

                    {/\*Image Area Start\*/}

                    <div className="form-group">

                        <label for="exampleInputEmail1">photo</label>

                        <input

                            type="file"

                            onChange={(e) => setimg(e.target.files[0])}

                            name="img"

                            accept="image/\*"

                            class="form-control"

                        />

                        {img !== "" && img !== undefined && img !== null ? (

                            <img

                                style={{ height: "180px" }}

                                // createObjectURL use for make image url

                                src={URL.createObjectURL(img)}

                                alt=""

                                className="upload-img"

                            />

                        ) : (

                            <>{img === "" && <p>Drag or drop content here</p>}</>

                        )}

                    </div>

                    {/\*Image area end\*/}

                    <button type="submit" class="btn btn-primary">{loading ?

                        <span className="spinner-border spinner-border-sm" role="status" aria-hidden="true"></span>

                        :

                        "Submit"}</button>

                </form>

            </div>

        </>

    )

}

export default Postformdata

What is React Toastify ?

React Toastify is a popular library for creating toast notifications in React applications.

Install React toastify : npm i react-toastify

Basic syntax of React toastify

import React from 'react';

import { ToastContainer, toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

function App(){

const notify = () => toast("Wow so easy!");

return (

<div>

<button onClick={notify}>Notify!</button>

<ToastContainer />

</div>

);

}

**33) Put data in Get API using raw post api :**

import axios from 'axios';

import React, { useState } from 'react';

// For Toast Message

import { ToastContainer, toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

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

const Addstudent = () => {

    // Decleare initial state

    const initialState = {

        name: "",

        email: "",

        phone: "",

        city: "",

        address: "",

        state: "",

        section: "",

        classes: ""

    };

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

    const [loading, setLoading] = useState(false); // State for loading indicator

    // const navigate = useNavigate();

    const [error, setError] = useState({}); // State for validation

    // Create validation function

    const validation = () => {

        let error = {}

        // For Name

        if (!user.name) {

            error.name = "Name is Required"

        } else if (user.name.length < 3) {

            error.name = "Name must be atleast 3 characters"

        }

        // For Email

        if (!user.email) {

            error.email = "Email is Required"

        } else if (!/^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)\*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/.test(user.email)) {

            error.email = "Email Should be xyz@gmail.com pattern"

        }

        // For Phone

        if (!user.phone) {

            error.phone = "Phone is Required"

        } else if (user.phone.length !== 10) {

            error.phone = "Phone number must be 10 characters"

        }

        // For City

        if (!user.city) {

            error.city = "City is Required"

        }

        // For Address

        if (!user.address) {

            error.address = "Address is Required"

        }

        // For State

        if (!user.state) {

            error.state = "State is Required"

        }

        // For Section

        if (!user.section) {

            error.section = "Section is Required"

        } else if (user.section.length < 3) {

            error.section = "Section must be atleast 3 characters"

        }

        // For Classes

        if (!user.classes) {

            error.classes = "Class is Required"

        }

        return error

    }

    const handleChange = (e) => {

        const { name, value } = e.target;

        setUser({ ...user, [name]: value });

        // Add Validation in Handle on change

        // For name

        if (name === 'name') {

            if (value.length === 0) {

                setError({ ...error, name: 'Name is Required' })

                setUser({ ...user, name: '' })

            } else {

                setError({ ...error, name: '' })

                setUser({ ...user, name: value })

            }

        }

        // For Email

        if (name === 'email') {

            if (value.length === 0) {

                setError({ ...error, email: 'Email is Required' })

                setUser({ ...user, email: '' })

            } else {

                setError({ ...error, email: '' })

                setUser({ ...user, email: value })

            }

        }

        // For phone

        if (name === 'phone') {

            if (value.length === 0) {

                setError({ ...error, phone: 'Phone is Required' })

                setUser({ ...user, phone: '' })

            } else {

                setError({ ...error, phone: '' })

                setUser({ ...user, phone: value })

            }

        }

        // For city

        if (name === 'city') {

            if (value.length === 0) {

                setError({ ...error, city: 'City is Required' })

                setUser({ ...user, city: '' })

            } else {

                setError({ ...error, city: '' })

                setUser({ ...user, city: value })

            }

        }

        // For address

        if (name === 'address') {

            if (value.length === 0) {

                setError({ ...error, address: 'Address is Required' })

                setUser({ ...user, address: '' })

            } else {

                setError({ ...error, address: '' })

                setUser({ ...user, address: value })

            }

        }

        // For state

        if (name === 'state') {

            if (value.length === 0) {

                setError({ ...error, state: 'State is Required' })

                setUser({ ...user, state: '' })

            } else {

                setError({ ...error, state: '' })

                setUser({ ...user, state: value })

            }

        }

        // For section

        if (name === 'section') {

            if (value.length === 0) {

                setError({ ...error, section: 'Section is Required' })

                setUser({ ...user, section: '' })

            } else {

                setError({ ...error, section: '' })

                setUser({ ...user, section: value })

            }

        }

        // For classes

        if (name === 'classes') {

            if (value.length === 0) {

                setError({ ...error, classes: 'Class is Required' })

                setUser({ ...user, classes: '' })

            } else {

                setError({ ...error, classes: '' })

                setUser({ ...user, classes: value })

            }

        }

    };

    // Handle for Submit

    const handleSubmit = async (e) => {

        e.preventDefault()

        setLoading(true) // Start Loading

        // For Error Handling

        let ErroList = validation()

        setError(validation())

        if (Object.keys(ErroList).length === 0) {

            const apiUrl = "https://tureappservar.onrender.com/student/create";

            // It is a Raw data so I am not using newFormdata()

            try {

                const response = await axios.post(apiUrl, user) // If it will formdata you will write formdata not user.

                console.log(response);

                toast.success(response?.data?.message)

                setLoading(false)

                setUser(initialState)

            } catch (error) {

                console.log(error);

                toast.error(error?.response?.data?.message)

                setLoading(false)

            }

        }else{

            setTimeout(() => {

                setLoading(false)

            }, 2000);

        }

    }

    return (

        <>

            <ToastContainer />

            {/\* <!-- contact section --> \*/}

            <section class="contact\_section layout\_padding">

                <div class="container">

                    <div class="heading\_container">

                        <h2>

                            Student Register

                        </h2>

                    </div>

                    <div class="row">

                        <div class="col-md-6">

                            <form onSubmit={handleSubmit} method='post'>

                                <div>

                                    <input type="text" name='name' placeholder="Name" onChange={handleChange}

                                        value={user.name} style={{ color: 'black' }} />

                                    <span style={{ display: 'block', color: 'red' }}>{error?.name}</span>

                                </div>

                                <div>

                                    <input type="email" name='email' placeholder="Email" onChange={handleChange} value={user.email} style={{ color: 'black' }} />

                                    <span style={{ display: 'block', color: 'red' }}>{error?.email}</span>

                                </div>

                                <div>

                                    <input type="text" name='phone' placeholder="Phone Number" onChange={handleChange} value={user.phone} style={{ color: 'black' }} />

                                    <span style={{ display: 'block', color: 'red' }}>{error?.phone}</span>

                                </div>

                                <div>

                                    <input type="text" name='city' placeholder="City" onChange={handleChange}

                                        value={user.city} style={{ color: 'black' }} />

                                    <span style={{ display: 'block', color: 'red' }}>{error?.city}</span>

                                </div>

                                <div>

                                    <input type="text" name='address' placeholder="Address" onChange={handleChange}

                                        value={user.address} style={{ color: 'black' }} />

                                    <span style={{ display: 'block', color: 'red' }}>{error?.address}</span>

                                </div>

                                <div>

                                    <input type="text" name='state' placeholder="State" onChange={handleChange}

                                        value={user.state} style={{ color: 'black' }} />

                                    <span style={{ display: 'block', color: 'red' }}>{error?.state}</span>

                                </div>

                                <div>

                                    <input type="text" name='section' placeholder="Section" onChange={handleChange}

                                        value={user.section} style={{ color: 'black' }} />

                                    <span style={{ display: 'block', color: 'red' }}>{error?.section}</span>

                                </div>

                                <div>

                                    <input type="text" name='classes' placeholder="Classes" onChange={handleChange}

                                        value={user.classes} style={{ color: 'black' }} />

                                    <span style={{ display: 'block', color: 'red' }}>{error?.classes}</span>

                                </div>

                                <div className="d-flex ">

                                    <button type='submit' className="btn btn-primary">

                                        {loading ?

                                            <span className="spinner-border spinner-border-sm" role="status" aria-hidden="true"></span>

                                            :

                                            "Submit"}

                                    </button>

                                </div>

                            </form>

                        </div>

                        <div class="col-md-6">

                            <div class="map\_container">

                                <div class="map">

                                    <div id="googleMap" style={{ width: '100%', height: '100%' }}></div>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

            </section>

            {/\* <!-- end contact section --> \*/}

        </>

    );

};

export default Addstudent;

For raw API, we will not use anything of formData.

**34) How to see Data is loaded or not ?**

Fillup form > Press Right click > Inspect > Network > Submit your form > If data is submitted then Submitted alert will be show on your screen or it is not submitted, Not submitted alert will be show > Select your data which is present in name field > Response > If Error will be make then Error message will be show , If Error will not be make then Your data will be show.

**35) What is useNavigate( ) in react ? and where and why we use useNavigate( ) ?**

`useNavigate()` is a hook provided by React Router that returns a function to programmatically navigate to different routes in a React application.

How to do it in code ?

Step1:

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

Step2:

const navigate=useNavigate() [Put it in hook section because it is a hook]

Step3:

navigate('/showstudent'); [Put this code where data is fetch just like try section]

**Note : If you add static successful data loaded message without navigation you can put this code**

 const adddata= axios.post(apiUrl, formdata)

     if(adddata){

        toast.success("data added successfully")

       }

**36) When I use (response.data[0]) in the times of useparams() details page ?**

First write console.log(response) under await > start npm > press right click and go to inspect > Console > Click on {data / object} which is present under warning > click on data >



0 is present in data in console that means we have to to do (response.data[0])

**37) What is CRUD in react ?**

CRUD in React refers to Create, Read, Update, and Delete operations for managing data within a React application.

**How to implement it in practical ?**

Necessary npm install for CRUD:

i) npm install react-router-dom

ii) npm i axios

**If API is not given :**

**iii) npm i json-server**

**iv) npm i concurrently**

**Go to package.json > scripts >**

"json-server": "json-server --watch src/Database/db.json --host 127.0.0.1 --port 3003",

"dev": "concurrently \"npm start\" \"npm run json-server\"",

**Put these two codes under the start : not into the start.**

Create a folder in src name will be Database > Put an API in this Database.

{

    "users": [

        {

            "name": "Raju kayal",

            "email": "kayalraju003@gmail.com",

            "phone": "08637000230",

            "password": "shdf1314242",

            "city": "kolkata",

            "state": "West Bengal",

            "dob": "2024-03-19",

            "id": 1

        }

    ]

}

**Run CRUD:**

Go to terminal > npm run dev > <http://localhost:3003/users>

put this link into view site information tab > press enter > You will see your API

Because Api is present in users array so I put /users in local host link

**These Area we will do If API is not Given.**

**App.js:**

import React from 'react'

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

import Nav from './Components/Nav'

import Adduser from './Components/Adduser'

import Showuser from './Components/Showuser'

import Edituser from './Components/Edituser'

const App = () => {

  return (

    <>

      <Router>

        <Nav/>

        <Routes>

            <Route path='/' element={<Adduser/>}/>

            <Route path='/showuser' element={<Showuser/>}/>

            <Route path='/edituser/:id' element={<Edituser/>}/>

        </Routes>

      </Router>

    </>

  )

}

export default App

**Note: I create a params between and Edituser page Edituser page is my Edit form page**

**Api.js**

import axios from "axios"

// Data is fetch in one time

const API\_URL = 'http://127.0.0.1:3003/users'

// For Add Data in Get API using Post API. (Use this function in Adduser.jsx file);

export const addUser = async (data) => {

    try {

        return await axios.post(API\_URL, data)

    }

    catch (error) {

        console.log('Error while calling addUser API', error.message)

    }

}

// For show Data in get API. (Use this function in Showuser.jsx file)

export const getUsers = async () => {

    try {

        return await axios.get(API\_URL)

    }

    catch (error) {

        console.log('Error while calling getUsers API', error.message)

    }

}

// This is make for edit data [For to show data in form after clicking edit button]

export const getUser = async (data) => {

    try {

        return await axios.get(`${API\_URL}/${data}`)

    }

    catch (error) {

        console.log('Error while calling getUser API', error.message)

    }

}

// This is actually part of edit data.

export const editUser = async (data, id) => {

    try {

        return await axios.put(`${API\_URL}/${id}`, data)

    }

    catch (error) {

        console.log('Error while calling editUser API', error.message)

    }

}

// This is make for delete data from table.

export const deleteUser = async (id) => {

    try {

        return await axios.delete(`${API\_URL}/${id}`)

    }

    catch (error) {

        console.log('Error while calling deleteUser API', error.message)

    }

}

**Note: I create a Folder name is Services. In Services folder I create a JS file Api.js . In this file I fetch Api url and keep it in different function. When I will needs any function then I will call that particular function.**

**Nav.jsx**

**Navbar is same as same as previous Routing navbar.**

**I take Adduser.jsx, Showuser.jsx, Edituser.jsx in Components folder**

**Adduser.jsx**

import React from 'react'

import { useState } from 'react'

import { addUser } from '../Services/Api' // Import addUser function

import { useNavigate } from 'react-router-dom' // useNavigate for Navigation

// For toast message

import { ToastContainer, toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

// Set initial values

const initialValues = {

    name: '',

    email: '',

    phone: '',

    password: "",

    city: "",

    state: "",

    dob: ""

}

const Adduser = () => {

    const [user, setUser] = useState(initialValues) // Make usestate for initialValues

    const [error, setError] = useState({}) // Make usestate for error message

    const [loading, setLoading] = useState(false); // Use state for Loading

    const navigate = useNavigate() // Make navigate to another page.

    // Create a Validation function which is to make validation on form field

    const validation = () => {

        let error = {} // I take error a blank object

        // If name is not present

        if (!user.name) {

            error.name = "Name is Required"

        } else if (user.name.length < 3) {

            error.name = "Name must be atleast 3 characters"

        }

        // If email is not present

        if (!user.email) {

            error.email = "Email is Required"

        } else if (

            !/^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)\*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/.test(user.email)

        ) {

            error.email = "Enter a valid Email"

        }

        // If phone is not present

        if (!user.phone) {

            error.phone = "Phone is Required";

        } else if (user.phone.length !== 10) {

            error.phone = "Phone number must be 10 characters";

        }

        // For password

        if (!user.password) {

            error.password = "Password is Required"

        } else if (user.password.length < 8) {

            error.password = "You must be entered a strong password"

        }

        // If phone is not present

        if (!user.city) {

            error.city = "City is Required"

        }

        // If state is not present

        if (!user.state) {

            error.state = "State is Required"

        }

        // If Date of birth is not present

        if (!user.dob) {

            error.dob = "Date of Birth is Required"

        }

        return error

    }

    // Make handle for changes

    const handleChange = (e) => {

        const { name, value } = e.target;

        setUser({ ...user, [name]: value });

        // In handle I set required field for all form field with name attribute

        if (name === "name") {

            if (value.length === 0) {

                setError({ ...error, name: "@Name is Required" })

                setUser({ ...user, name: "" })

            } else {

                setError({ ...error, name: "" })

                setUser({ ...user, name: value })

            }

        }

        if (name === "email") {

            if (value.length === 0) {

                setError({ ...error, email: "@Email is Required" })

                setUser({ ...user, email: "" })

            } else {

                setError({ ...error, email: "" })

                setUser({ ...user, email: value })

            }

        }

        if (name === "phone") {

            if (value.length === 0) {

                setError({ ...error, phone: "@Phone is Required" })

                setUser({ ...user, phone: "" })

            } else {

                setError({ ...error, phone: "" })

                setUser({ ...user, phone: value })

            }

        }

        if (name === "password") {

            if (value.length === 0) {

                setError({ ...error, password: "@Password Line One is required" })

                setUser({ ...user, password: "" })

            } else {

                setError({ ...error, password: "" })

                setUser({ ...user, password: value })

            }

        }

        if (name === "city") {

            if (value.length === 0) {

                setError({ ...error, city: "@City is Required" })

                setUser({ ...user, city: "" })

            } else {

                setError({ ...error, city: "" })

                setUser({ ...user, city: value })

            }

        }

        if (name === "state") {

            if (value.length === 0) {

                setError({ ...error, state: "@State is Required" })

                setUser({ ...user, state: "" })

            } else {

                setError({ ...error, state: "" })

                setUser({ ...user, state: value })

            }

        }

        if (name === "dob") {

            if (value.length === 0) {

                setError({ ...error, dob: "@Date of Birth is Required" })

                setUser({ ...user, dob: "" })

            } else {

                setError({ ...error, dob: "" })

                setUser({ ...user, dob: value })

            }

        }

    };

    // Make handle for submit

    const handleSubmit = async (e) => {

        e.preventDefault(); // For to stop reloading the page

        setLoading(true) // After clicking submit button Loading will be start

        let ErrorList = validation() // Call validation function in ErrorList variable

        setError(validation())

        // Validiation for Submit

        if (Object.keys(ErrorList).length === 0) {

            await addUser(user) // call addUser function

            .then((response) =>{

                console.log(response);

                setLoading(false) // Loading will be stop after loading data

                toast.success("User added successfully")

                setTimeout(() => {

                    navigate('/showuser')

                }, 2000);

            })

            .catch((error) =>{

                console.log(error);

                setLoading(false) // Loading will be stop after not loading data

                toast.error("User is not added")

            })

        } else {

            setTimeout(() => {

                setLoading(false);

            }, 200);

        }

    };

    return (

        <>

            <ToastContainer style={{ top: '10%', left: '50%', transform: 'translate(-50%, -50%)', alignItems: 'center', textAlign: 'center' }} />

            <section className="contact\_section layout\_padding" style={{ backgroundColor: '#f8f9fa', padding: '50px 0' }}>

                <div className="container">

                    <div className="heading\_container text-center">

                        <h2 style={{ color: '#333' }}>Registration</h2>

                    </div>

                    <div className="row">

                        <div className="col-md-6">

                            <form onSubmit={handleSubmit}>

                                <div className="form-group">

                                    <input

                                        type="text"

                                        name="name"

                                        value={user.name}

                                        onChange={handleChange}

                                        placeholder="Name"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', display: 'block' }}> {error.name} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="email"

                                        name="email"

                                        value={user.email}

                                        onChange={handleChange}

                                        placeholder="Email"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', display: 'block' }}> {error.email} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="text"

                                        name="phone"

                                        value={user.phone}

                                        onChange={handleChange}

                                        placeholder="Phone Number"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', display: 'block' }}> {error.phone} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="password"

                                        name="password"

                                        value={user.password}

                                        onChange={handleChange}

                                        placeholder="Password"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', display: 'block' }}> {error.password} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="text"

                                        name="city"

                                        value={user.city}

                                        onChange={handleChange}

                                        placeholder="City"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', display: 'block' }}> {error.city} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="text"

                                        name="state"

                                        value={user.state}

                                        onChange={handleChange}

                                        placeholder="State"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', display: 'block' }}> {error.state} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="date"

                                        name="dob"

                                        value={user.dob}

                                        onChange={handleChange}

                                        placeholder="Date of birth"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', display: 'block' }}> {error.dob} </span>

                                </div>

                                <button type="submit" className="btn btn-primary" style={{ backgroundColor: '#007bff', borderColor: '#007bff', color: '#fff', borderRadius: '5px', padding: '10px 20px', marginTop: '10px' }}>

                                    {loading ?

                                        <div className="spinner-border" role="status">

                                            <span className="sr-only">Loading...</span>

                                        </div>

                                        :

                                        'Submit'}

                                </button>

                            </form>

                        </div>

                    </div>

                </div>

            </section>

        </>

    )

}

export default Adduser

**Why we use preventDefault in handlesumbit() ?**

**preventDefault()** is used in **handleSubmit()** to stop the default behavior of form submission, which involves reloading the page.

**How to add Form field validation in Form in Adduser.jsx ?**

**Step1: Create a blank object state**

const [error, setError] = useState({}) // Make usestate for error message

**Step2: Create a validation function**

// Create a Validation function which is to make validation on form field

    const validation = () => {

        let error = {} // I take error a blank object

        // If name is not present

        if (!user.name) {

            error.name = "Name is Required"

        } else if (user.name.length < 3) {

            error.name = "Name must be atleast 3 characters"

        }

        // If email is not present

        if (!user.email) {

            error.email = "Email is Required"

        } else if (

            !/^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)\*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/.test(user.email)

        ) {

            error.email = "Enter a valid Email"

        }

        // If phone is not present

        if (!user.phone) {

            error.phone = "Phone is Required"

        } else if (user.phone.length !== 10) {

            error.phone = "Phone number must be 10 characters"

        }

        // For password

        if (!user.password) {

            error.password = "Password is Required"

        } else if (user.password.length < 8) {

            error.password = "You must be entered a strong password"

        }

        // If phone is not present

        if (!user.city) {

            error.city = "City is Required"

        }

        // If state is not present

        if (!user.state) {

            error.state = "State is Required"

        }

        // If Date of birth is not present

        if (!user.dob) {

            error.dob = "Date of Birth is Required"

        }

        return error

    }

**Step2: Put for validation in handle change area**

// Make handle for changes

    const handleChange = (e) => {

        const { name, value } = e.target;

        setUser({ ...user, [name]: value });

        // In handle I set required field for all form field with name attribute

        if (name === "name") {

            if (value.length === 0) {

                setError({ ...error, name: "@Name is Required" })

                setUser({ ...user, name: "" })

            } else {

                setError({ ...error, name: "" })

                setUser({ ...user, name: value })

            }

        }

        if (name === "email") {

            if (value.length === 0) {

                setError({ ...error, email: "@Email is Required" })

                setUser({ ...user, email: "" })

            } else {

                setError({ ...error, email: "" })

                setUser({ ...user, email: value })

            }

        }

        if (name === "phone") {

            if (value.length === 0) {

                setError({ ...error, phone: "@Phone is Required" })

                setUser({ ...user, phone: "" })

            } else {

                setError({ ...error, phone: "" })

                setUser({ ...user, phone: value })

            }

        }

        if (name === "password") {

            if (value.length === 0) {

                setError({ ...error, password: "@Password Line One is required" })

                setUser({ ...user, password: "" })

            } else {

                setError({ ...error, password: "" })

                setUser({ ...user, password: value })

            }

        }

        if (name === "city") {

            if (value.length === 0) {

                setError({ ...error, city: "@City is Required" })

                setUser({ ...user, city: "" })

            } else {

                setError({ ...error, city: "" })

                setUser({ ...user, city: value })

            }

        }

        if (name === "state") {

            if (value.length === 0) {

                setError({ ...error, state: "@State is Required" })

                setUser({ ...user, state: "" })

            } else {

                setError({ ...error, state: "" })

                setUser({ ...user, state: value })

            }

        }

        if (name === "dob") {

            if (value.length === 0) {

                setError({ ...error, dob: "@Date of Birth is Required" })

                setUser({ ...user, dob: "" })

            } else {

                setError({ ...error, dob: "" })

                setUser({ ...user, dob: value })

            }

        }

    };

**Step3: Make Handle for Submit**

// Make handle for submit

    const handleSubmit = async (e) => {

        e.preventDefault(); // For to stop reloading the page

        setLoading(true) // After clicking submit button Loading will be start

        let ErrorList = validation() // Call validation function in ErrorList variable

        setError(validation())

        if (Object.keys(ErrorList).length === 0) {

            await addUser(user) // call addUser function

            //Navigate to showuser page

            navigate('/showuser')

            //Loading will be stop after submitting the data

            setLoading(false)

        } else {

            setTimeout(() => {

                setLoading(false);

            }, 200);

        }

    };

**Note: For to add form validation in Edituser.jsx page you have to add** if (Object.keys(ErrorList).length === 0) **this code on handle click area. Code is present in Edituser.jsx go and check.**

Step3:Put Validation Field under input of form.

<span style={{ color: 'red', display: 'block' }}> {error.name} </span>

**Showuser.jsx**

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

import { getUsers } from '../Services/Api'; // Import getUsers() function

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

import { deleteUser } from '../Services/Api'; // Function for delete

import { ToastContainer, toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

const Showuser = () => {

    const [user, setuser] = useState([]);

    const getdata = async () => {

        const response = await getUsers() // Call getUsers() function

        setuser(response.data)

    }

    useEffect(() => {

        getdata() // Call getdata()function

    }, [])

    // Create a function for delete

const deleteUserData = async (id) => {

        await deleteUser(id)

            .then((response) => {

                console.log(response);

                getdata() // Call getdata()function

                toast.warn("User Deleted successfully");

            })

            .catch((error) =>{

                console.log(error);

                toast.error("User is not Deleted")

            })

    }

    console.log('hd', user);

    return (

        <>

            <ToastContainer style={{ top: '10%', left: '50%', transform: 'translate(-50%, -50%)', alignItems: 'center', textAlign: 'center' }} />

            <div className="container-fluid mt-5">

                <table className="table table-bordered table-striped">

                    <thead className="thead-dark">

                        <tr>

                            <th>Name</th>

                            <th>Email</th>

                            <th>Phone</th>

                            <th>Password</th>

                            <th>City</th>

                            <th>State</th>

                            <th>DOB</th>

                            <th>Edit</th>

                            <th>Delete</th>

                        </tr>

                    </thead>

                    {user?.slice(0, user.length).reverse().map((value) => {

                        return (

                            <>

                                <tbody key={value.id}>

                                    <tr>

                                        <td>{value.name}</td>

                                        <td>{value.email}</td>

                                        <td>{value.phone}</td>

                                        <td>{value.password}</td>

                                        <td>{value.city}</td>

                                        <td>{value.state}</td>

                                        <td>{value.dob}</td>

                                        <td><Link to={`/edituser/${value.id}`}><button type="button" class="btn btn-primary">Edit</button></Link></td>

                                        <td><button type="button" class="btn btn-danger" onClick={() => deleteUserData(value.id)}>Delete</button></td>

                                    </tr>

                                </tbody>

                            </>

                        )

                    })}

                </table>

            </div>

        </>

    )

}

export default Showuser

**Edituser.jsx**

// Note: Here, the validation() and postUserData() is not really required as we're updating few changes while other credentials are pre-filled by user and not kept blank.

import React from 'react'

import { useState, useEffect } from 'react'

import { useNavigate } from 'react-router-dom'

import { useParams } from 'react-router-dom'

import { editUser, getUser } from '../Services/Api' // Import two functions which I create for edit

// For toast message

import { ToastContainer, toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

// Set initial values

const initialValues = {

    name: '',

    email: '',

    phone: '',

    password: "",

    city: "",

    state: "",

    dob: ""

}

const Edituser = () => {

    const [user, setUser] = useState(initialValues) // Usestate for initial state

    const [error, setError] = useState({}) // Use state for validation

    const [loading, setLoading] = useState(false); // Use state for loading

    const navigate = useNavigate() // Use navigate for another page

    const { id } = useParams() // Create use params

    // Create validation function

    const validation = () => {

        let error = {}

        if (!user.name) {

            error.name = "Name is Required"

        }

        if (!user.email) {

            error.email = "Email is Required"

        } else if (

            !/^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)\*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/.test(user.email)

        ) {

            error.email = "Enter a valid Email"

        }

        if (!user.phone) {

            error.phone = "Phone is Required";

        } else if (user.phone.length !== 10) {

            error.phone = "Phone number must be 10 characters";

        }

        if (!user.password) {

            error.password = "Password is Required"

        } else if (user.password.length < 8) {

            error.password = "Please Enter Valid password"

        }

        if (!user.city) {

            error.city = "City is Required"

        }

        if (!user.state) {

            error.state = "State is Required"

        }

        if (!user.dob) {

            error.dob = "Date of Birth is Required"

        }

        return error

    }

    // Make handle for changes

    const handleChange = (e) => {

        const { name, value } = e.target;

        setUser({ ...user, [name]: value });

        // In handle I set required field for all form field

        if (name === "name") {

            if (value.length === 0) {

                setError({ ...error, name: "@Name is Required" })

                setUser({ ...user, name: "" })

            } else {

                setError({ ...error, name: "" })

                setUser({ ...user, name: value })

            }

        }

        if (name === "email") {

            if (value.length === 0) {

                setError({ ...error, email: "@Email is Required" })

                setUser({ ...user, email: "" })

            } else {

                setError({ ...error, email: "" })

                setUser({ ...user, email: value })

            }

        }

        if (name === "phone") {

            if (value.length === 0) {

                setError({ ...error, phone: "@Phone is Required" })

                setUser({ ...user, phone: "" })

            } else {

                setError({ ...error, phone: "" })

                setUser({ ...user, phone: value })

            }

        }

        if (name === "password") {

            if (value.length === 0) {

                setError({ ...error, password: "@Password Line One is required" })

                setUser({ ...user, password: "" })

            } else {

                setError({ ...error, password: "" })

                setUser({ ...user, password: value })

            }

        }

        if (name === "city") {

            if (value.length === 0) {

                setError({ ...error, city: "@City is Required" })

                setUser({ ...user, city: "" })

            } else {

                setError({ ...error, city: "" })

                setUser({ ...user, city: value })

            }

        }

        if (name === "state") {

            if (value.length === 0) {

                setError({ ...error, state: "@State is Required" })

                setUser({ ...user, state: "" })

            } else {

                setError({ ...error, state: "" })

                setUser({ ...user, state: value })

            }

        }

        if (name === "dob") {

            if (value.length === 0) {

                setError({ ...error, dob: "@Date of Birth is Required" })

                setUser({ ...user, dob: "" })

            } else {

                setError({ ...error, dob: "" })

                setUser({ ...user, dob: value })

            }

        }

    };

    const getData = async () => {

        let response = await getUser(id) // Call getUser()function for edit

        setUser(response?.data)

        console.log(response);

    }

    useEffect(() => {

        getData() // Call getData()function

    }, [])

    const handleOnSubmit = async (e) => {

        e.preventDefault()

        setLoading(true);

        let ErrorList = validation()

        setError(validation())

       // Validation for Submit

        if (Object.keys(ErrorList).length === 0) {

            let reg = {

                name: user.name,

                email: user.email,

                phone: user.phone,

                password: user.address,

                city: user.city,

                state: user.class,

                dob: user.dob

            }

            console.log(reg)

        }

    }

    // This function is make to click on edit button

    const handleOnClick = async () => {

        let ErrorList = validation()

        setError(validation())

        // This is make for use validation

        if (Object.keys(ErrorList).length === 0) {

            await editUser(user, id) // call editUser()function for edit

                .then((response) => {

                    console.log(response);

                    toast.success("User edited successfully");

                    setTimeout(() => {

                        setLoading(false);

                        navigate('/showuser')

                    }, 2000);

                })

                .catch((error) => {

                    console.log(error);

                    toast.error("User is not edit");

                    setLoading(false);

                })

        } else {

            setTimeout(() => {

                setLoading(false)

            }, 1000);

        }

    }

    return (

        <>

            <ToastContainer style={{ top: '10%', left: '50%', transform: 'translate(-50%, -50%)', alignItems: 'center', textAlign: 'center' }} />

            <section className="contact\_section layout\_padding" style={{ backgroundColor: '#f8f9fa', padding: '50px 0' }}>

                <div className="container">

                    <div className="heading\_container text-center">

                        <h2 style={{ color: '#333' }}>Edit User</h2>

                    </div>

                    <div className="row">

                        <div className="col-md-6">

                            <form onSubmit={handleOnSubmit}>

                                <div className="form-group">

                                    <input

                                        type="text"

                                        name="name"

                                        value={user.name}

                                        onChange={handleChange}

                                        placeholder="Name"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', marginLeft: '24px', display: 'block' }}> {error.name} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="email"

                                        name="email"

                                        value={user.email}

                                        onChange={handleChange}

                                        placeholder="Email"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', marginLeft: '24px', display: 'block' }}> {error.email} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="text"

                                        name="phone"

                                        value={user.phone}

                                        onChange={handleChange}

                                        placeholder="Phone Number"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', marginLeft: '24px', display: 'block' }}> {error.phone} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="password"

                                        name="password"

                                        value={user.password}

                                        onChange={handleChange}

                                        placeholder="Password"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', marginLeft: '24px', display: 'block' }}> {error.password} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="text"

                                        name="city"

                                        value={user.city}

                                        onChange={handleChange}

                                        placeholder="City"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', marginLeft: '24px', display: 'block' }}> {error.city} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="text"

                                        name="state"

                                        value={user.state}

                                        onChange={handleChange}

                                        placeholder="State"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', marginLeft: '24px', display: 'block' }}> {error.state} </span>

                                </div>

                                <div className="form-group">

                                    <input

                                        type="date"

                                        name="dob"

                                        value={user.dob}

                                        onChange={handleChange}

                                        placeholder="Date of birth"

                                        className="form-control"

                                        style={{ color: 'black', border: '1px solid #ccc', borderRadius: '5px', padding: '10px', marginBottom: '15px' }}

                                    />

                                    <span style={{ color: 'red', marginLeft: '24px', display: 'block' }}> {error.dob} </span>

                                </div>

                                <button type="submit" className="btn btn-primary" onClick={handleOnClick} style={{ backgroundColor: '#007bff', borderColor: '#007bff', color: '#fff', borderRadius: '5px', padding: '10px 20px', marginTop: '10px' }}>

                                    {loading ?

                                        <div className="spinner-border" role="status">

                                            <span className="sr-only">Loading...</span>

                                        </div>

                                        :

                                        'Submit'}

                                </button>

                            </form>

                        </div>

                        <div className="col-md-6">

                            <div className="map\_container">

                                <div className="map">

                                    <div id="googleMap" style={{ width: '100%', height: '100%' }}></div>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

            </section>

        </>

    )

}

export default Edituser

How to Show Toast Notification on All user page after performing Delete or Edit ?

Step1:

// Import Toast In App.js for global Toast

import { ToastContainer } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

Step2:

<Router>

        <ToastContainer /> // Keep ToastContainer in this area

        <Nav />

        <Routes>

          <Route path='/' element={<Addstudent />} />

          <Route path='/showstudent' element={<Showstudent />} />

          <Route path='/editstudent/:id' element={<Editstudent />} />

        </Routes>

      </Router>

Step3: import { toast } from 'react-toastify'; // Import this in All user page

Step4:

.then((response) => {

                    console.log(response);

                    setTimeout(() => {

                        // Show success toast

                        toast.success(response.data.msg); // Keep toast in SetTimeout

                    },1000);

                    // Reset form values

                    setStudent(initialValues);

                    navigate('/showstudent') // keep navigate without setTimeout

                    // Stop loading

                    setLoading(false);

                })

                .catch((error) => {

                    console.error('Error adding user:', error);

                    // Handle error if necessary

                    setLoading(false);

                });

**CRUD Done**

**38) What is Context in React ?**

In React, context provides a way to pass data through the component tree without having to pass props down manually at every level.

How many type of step are present in React context ?

In React context, there are typically three steps involved: creating the context, providing the context value, and consuming the context value.

How to implement it in practical ?

Step1 : Create a file name will be Createcontext.jsx

Createcontext.jsx

import { createContext } from "react";

const createcontext = createContext()

export default createcontext;

Step2: Create an another file name will be Getcontext.jsx

import React from 'react'

// Import createcontext function

import createcontext from './Createcontext'

const Getcontext = ({children}) => {

    const user={

        name: "Mousom",

        phone: '9903419235',

        email:"mousom@gmail.com",

        city:"Kolkata"

    }

  return (

    <>

      <createcontext.Provider value={user}>

        {children}

      </createcontext.Provider>

    </>

  )

}

export default Getcontext

Step2: Import createcontext function in User.jsx and Student.jsx

User.jsx

import React, { useContext } from 'react'

import createcontex from './Createcontext'

const User = () => {

    const mydata = useContext(createcontex)

    return (

        <div>

            <h1>User Component</h1>

            <h3>{mydata.name}</h3>

            <h3>{mydata.email}</h3>

            <h3>{mydata.city}</h3>

        </div>

    )

}

export default User

Student.jsx

import React, { useContext } from 'react'

import createcontex from './Createcontext'

const Student = () => {

    const mydata = useContext(createcontex)

    return (

        <div>

            <h1>Student Component</h1>

            <h3>{mydata.name}</h3>

            <h3>{mydata.email}</h3>

            <h3>{mydata.city}</h3>

        </div>

    )

}

export default Student

Step3: Import Getcontext.jsx component in App.js . All components will be stay in <Getcontext> including Router.

App.js

import Getcontext from './Context/Getcontext';

import Nav from './Context/Nav'

import User from './Context/User';

import Student from './Context/Student';

<Getcontext>

        <Router>

          <Nav />

          <Routes>

            <Route path='/' element={<User />} />

            <Route path='/student' element={<Student />} />

          </Routes>

        </Router>

      </Getcontext>

What is children in Context?

"children" can be passed as a prop in React, allowing any nested elements or other components to be passed down within a component or as their props, although it's commonly considered a default prop, it's extremely useful for component composition or inheritance.

What Provider in Context?

In React context, ".Provider" is a component used to wrap parts of the component tree to provide the context values to its descendants.

**39) How to Add MS office style in Form Input message using React Quill ?**

**Step1: npm install react-quill**

**Step2:**

//For MS office style

import ReactQuill from 'react-quill';

import 'react-quill/dist/quill.snow.css';

**Step3: Create a Handle function for React Quill**

// This function is make of React quill

    const handleEditorChange = (content) => {

        setBlog({ ...blog, yourblog: content });

    };

**blog is your use state piller.**

**Step4:**

<ReactQuill

                                    theme="snow"

                                    name="yourblog"

                                    value={blog.yourblog}

                                    onChange={handleEditorChange}

                                    placeholder="Write your blog here..."

                                    style={{ height: '300px' }}

                                    required

                                />

40) How to use React Spinner Loader ?

Step1: npm i react-loader-spinner

Step2: Search in Google react-loader-spinner > Go to npm website > Spinners > Choose a spinner.

Step3: Create a jsx file in Common folder name will be Loading.jsx

import React from 'react'

import { Vortex } from 'react-loader-spinner'

const Loading = () => {

    return (

        <>

            <Vortex

                visible={true}

                height="200"

                width="200"

                ariaLabel="vortex-loading"

                wrapperStyle={{}}

                wrapperClass="vortex-wrapper"

                colors={['red', 'green', 'blue', 'yellow', 'orange', 'purple']}

            />

        </>

    )

}

export default Loading

In this case I chose Vortex. You can chose anything. Loader components code will be present in npm website.

Step4: Where you want to show loading ...

i) import Loading from './Loading' // Import loading function

ii) setLoading(false) // It will be false after Loading data

iii)  if (loading) {

        return (

            <>

                <h1 style={{ position: 'absolute', top: '50%', left: '50%', transform: 'translate(-50%, -50%)'}}><Loading /></h1>

            </>

        )

    }

41) How to show image in get data using image API link ?

Blog.jsx

<img src={`https://restapinodejs.onrender.com/api/blog/image/$ {item.\_id}`} alt="" />

Details.jsx

Write ${id} in the place of ${item.\_id}

42) What is Private Routing and How to apply it in practical ?

Private routing in a website refers to the process of restricting access to certain pages or content based on user authentication or authorization, ensuring that only authorized users can view or interact with specific parts of the site.

How to use it in practical ?

**First Basic Step**

Step1: Create a File in your src, name will be Layout.jsx

Layout.jsx

import React from 'react'

import Nav from '../Common/Nav'

import Footer from './Footer'

const Layout = ({children}) => {

  return (

    <>

      <Nav/>

      <main>

        {children}

      </main>

      <Footer/>

    </>

  )

}

export default Layout

Keep <Nav/> in the top of <main> and keep <Footer/> in the button of <main>

Step 2 : Nav Bar and Footer ছাড়া যতগুলি পেজ তুমি বানাবে সেই পেজ গুলির Main Function এর রিটার্ন এরিয়া, Layout কম্পোনেন্ট দিয়ে মুড়ে দাও Just like this :

import React from 'react'

import Layout from './Layout'

const Home = () => {

    return (

        <>

            <Layout>

                <h1>Welcome to Home Page</h1>

            </Layout>

        </>

    )

}

export default Home

Step 3 : Make a New style Routing in App.js file

App.js

import React from 'react'

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

import Register from './Pages/Register'

import Login from './Pages/Login'

import Home from './Pages/Home'

const App = () => {

   // Public Route For Public Page

    const public\_route = [

        {

            path:"/login",

            component:<Login/>

        },

        {

            path:"/register",

            component:<Register/>

        }

    ]

    // Private Route For Private Page

    const private\_route = [

        {

            path:"/",

            component:<Home/>

        }

    ]

    return (

        <>

            <Router>

                <Routes>

                    {

                        public\_route?.map((value) =>{

                            return(

                                <>

                               <Route path={value.path} element={value.component}/>

                                </>

                            )

                        })

                    }

                    {

                        private\_route?.map((value) =>{

                            return(

                                <>

                                <Route path={value.path} element={value.component}/>

                                </>

                            )

                        })

                    }

                </Routes>

            </Router>

        </>

    )

}

export default App

**Note:** These 3steps are the very first step for Private Routing it was very basic now I move advance area.

**Next Advance Step**

**Step1:** Create Login.jsx, Register.jsx and others components as same as previous.

**Step2:** Create a file Authcontext.jsx in your src.

**Authcontext.jsx**

import { useState,useContext, createContext, useEffect } from "react";

const AuthContext = createContext();

const AuthProvider = ({ children }) => {

  const [auth, setAuth] = useState({

    user: null,

    token: "",

  });

  useEffect(()=>{

        const data=localStorage.getItem('auth')

        if(data){

            const parseData=JSON.parse(data)

            setAuth({

                ...auth,

                user:parseData.user,

                token:parseData.token

            })

        }

  },[])

  return (

    <AuthContext.Provider value={[auth, setAuth]}>

      {children}

    </AuthContext.Provider>

  );

};

// Custom hook

const useAuth = () => useContext(AuthContext);

export { useAuth, AuthProvider };

**Note:** AuthProvider কে আমরা পাঠাবো index .js এ এবং useauth কে পাঠাবো Login.jsx এএবং Nav.jsx এ

**Index.js**

**import { AuthProvider } from './Context/Authcontext';**

**// Cover with <AuthProvider> function**

  <AuthProvider>

    <React.StrictMode>

      <App />

    </React.StrictMode>

  </AuthProvider>

**Login.jsx**

import React, { useState } from 'react'

import Layout from './Layout'

import { toast } from 'react-toastify';

import { useNavigate } from 'react-router-dom'

import axios from 'axios'

import { useAuth } from '../Context/Authcontext'; // Import useAuth

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

const Login = () => {

    const [email, setEmail] = useState('');

    const [password, setPassword] = useState('')

    const [auth, setAuth] = useAuth() // Custom Hook

    const [loading, setloading] = useState(false)

    const navigate = useNavigate()

    const handlelogin = async (e) => {

        e.preventDefault()

        setloading(true)

        try {

            const response = await axios.post("https://ecommerce-2k0v.onrender.com/api/auth/login", { email, password });

           // This step is very import for auth. success === true can be status === 200 or others for another api

            if (response && response?.data.success === true) {

                toast.success(response?.data?.message);

                // Auth start

                setAuth({

                    ...auth,

                    user: response.data.user,

                    token: response.data.token

                });

                localStorage.setItem("auth", JSON.stringify(response?.data))

                // Auth end

                navigate('/') // Navigate Home Page

                setloading(false)

            } else {

                toast.error(response?.data?.message);

                setloading(false)

            }

        } catch (error) {

            console.log(error)

            toast.error(error.response.data.message)

            setloading(false)

        }

    }

    return (

        <>

            <Layout>

                <div className='container-fluid vh-100'>

                    <div className='mx-auto mt-5 p-4 border border-success rounded-3' style={{ width: "400px", boxShadow: "0px 4px 8px green" }}>

                        <form onSubmit={handlelogin}>

                            <h4 className='text-center'>Login Page</h4>

                            <div class="mb-3">

                                <label for="exampleInputEmail1" class="form-label">Email address</label>

                                <input type="email" value={email} onChange={(e) => setEmail(e.target.value)} class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" />

                            </div>

                            <div class="mb-3">

                                <label for="exampleInputPassword1" class="form-label">Password</label>

                                <input type="password" value={password} onChange={(e) => setPassword(e.target.value)} class="form-control" id="exampleInputPassword1" />

                            </div>

                            <div className='mb-3'>

                                <Link to="/forgetpassword">Forget password</Link>

                            </div>

                            <button type="submit" class="btn btn-success">{loading ? <div class="spinner-border" role="status">

                                <span class="sr-only">Loading...</span>

                            </div> : "Login"}</button>

                        </form>

                    </div>

                </div>

            </Layout>

        </>

    )

}

export default Login

These are three api for Login,Register and Forgetpassword

const registerapi = '<https://ecommerce-2k0v.onrender.com/api/auth/register>'

const loginapi = '<https://ecommerce-2k0v.onrender.com/api/auth/login>'

const forgetapi = "<https://ecommerce-2k0v.onrender.com/api/auth/forgot-password>"

Register.jsx:

import React, { useState } from 'react'

import Layout from './Layout';

import axios from 'axios';

import { toast } from 'react-toastify';

import { useNavigate } from "react-router-dom";

const Register = () => {

    // create Hooks for all form field

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

    const [email, setEmail] = useState("");

    const [password, setPassword] = useState("");

    const [phone, setPhone] = useState("");

    const [address, setAddress] = useState("");

    const [answer, setAnswer] = useState("");

    const [loading, setLoading] = useState(false);

    const navigate = useNavigate(); // Navigate another page

    const handleOnSubmit = async (e) => {

        e.preventDefault();

        setLoading(true); // Loading will be start

        try {

            const response = await axios.post('https://ecommerce-2k0v.onrender.com/api/auth/register', {

                name,

                email,

                password,

                phone,

                address,

                answer,

            });

            // This step is must for Private routing

            if (response && response?.data?.success === true) {

                toast.success(response?.data?.message)

                setLoading(false)

                navigate("/login");

            } else {

                toast.error(response?.data?.message)

                setLoading(false);

            }

        } catch (error) {

            // Handle any errors that occurred during the axios request

            console.error('Error occurred:', error);

            toast.error(error?.response?.data?.message)

            setLoading(false)

        }

    };

    return (

        <>

            <Layout>

                <div className='container-fluid vh-100'>

                    <div className='mx-auto mt-5 p-4 border border-info rounded-3' style={{ width: "400px", boxShadow: "0px 4px 8px blue" }}>

                        <form onSubmit={handleOnSubmit}>

                            <div class="mb-3">

                                <label for="exampleInputEmail1" class="form-label">Name</label>

                                <input type="text" class="form-control" value={name} onChange={(e) => setName(e.target.value)} id="" aria-describedby="emailHelp" />

                            </div>

                            <div class="mb-3">

                                <label for="exampleInputEmail1" class="form-label">Email</label>

                                <input type="email" class="form-control" value={email} onChange={(e) => setEmail(e.target.value)} id="exampleInputEmail1" aria-describedby="emailHelp" />

                            </div>

                            <div class="mb-3">

                                <label for="exampleInputEmail1" class="form-label">phone</label>

                                <input type="tel" class="form-control" value={phone} onChange={(e) => setPhone(e.target.value)} id="exampleInputEmail1" aria-describedby="emailHelp" />

                            </div>

                            <div class="mb-3">

                                <label for="exampleInputEmail1" class="form-label">Address</label>

                                <input type="text" class="form-control" value={address} onChange={(e) => setAddress(e.target.value)} id="exampleInputEmail1" aria-describedby="emailHelp" />

                            </div>

                            <div class="mb-3">

                                <label for="exampleInputPassword1" class="form-label">Password</label>

                                <input type="password" class="form-control" value={password} onChange={(e) => setPassword(e.target.value)} id="exampleInputPassword1" />

                            </div>

                            <div class="mb-3">

                                <label for="exampleInputPassword1" class="form-label">Answer</label>

                                <input type="text" class="form-control" value={answer} onChange={(e) => setAnswer(e.target.value)} id="exampleInputPassword1" />

                            </div>

                            <button type="submit" class="btn btn-primary">

                                {loading ? <div class="spinner-border" role="status">

                                    <span class="sr-only">Loading...</span>

                                </div> : 'Register'}

                            </button>

                        </form>

                    </div>

                </div>

            </Layout>

        </>

    )

}

export default Register

**Forgetpassword.js**

import React, { useState } from 'react';

import Layout from './Layout';

import { toast } from 'react-toastify';

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

import axios from 'axios';

const Forgetpassword = () => {

    const [email, setEmail] = useState('');

    const [newPassword, setPassword] = useState('');

    const [answer, setAnswer] = useState('');

    const [loading, setLoading] = useState(false);

    const navigate = useNavigate();

    const handleForget = async (e) => {

        e.preventDefault();

        try {

            setLoading(true);

            const response = await axios.post("https://ecommerce-2k0v.onrender.com/api/auth/forgot-password", { email, newPassword, answer });

            // This step is necessary for private routing

            if (response.data.success === true) {

                toast.success(response.data.message);

                navigate('/login');

                setLoading(false)

            } else {

                toast.error(response?.data?.message);

                setLoading(false)

            }

        } catch (error) {

            console.error(error);

            toast.error(error.response.data.message);

            setLoading(false)

        }

    };

    return (

        <Layout>

            <div className='container-fluid vh-100'>

                <div className='mx-auto mt-5 p-4 border border-success rounded-3' style={{ width: "400px", boxShadow: "0px 4px 8px green" }}>

                    <form onSubmit={handleForget}>

                        <h4 className='text-center'>Forget Password</h4>

                        <div className="mb-3">

                            <label htmlFor="exampleInputEmail1" className="form-label">Email address</label>

                            <input type="email" value={email} onChange={(e) => setEmail(e.target.value)} className="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" />

                        </div>

                        <div className="mb-3">

                            <label htmlFor="exampleInputPassword1" className="form-label">Password</label>

                            <input type="password" value={newPassword} onChange={(e) => setPassword(e.target.value)} className="form-control" id="exampleInputPassword1" />

                        </div>

                        <div className="mb-3">

                            <label htmlFor="exampleInputAnswer1" className="form-label">Answer</label>

                            <input type="text" value={answer} onChange={(e) => setAnswer(e.target.value)} className="form-control" id="exampleInputAnswer1" />

                        </div>

                        <button type="submit" className="btn btn-success">{loading ? <div class="spinner-border" role="status">

                            <span class="sr-only">Loading...</span>

                        </div> : "Submit"}</button>

                    </form>

                </div>

            </div>

        </Layout>

    );

};

export default Forgetpassword;

**App.js**

import React from 'react'

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

import Register from './Pages/Register'

import Login from './Pages/Login'

import Home from './Pages/Home'

import {Navigate} from 'react-router-dom'

import 'react-toastify/dist/ReactToastify.css';

import { ToastContainer} from 'react-toastify';

import Forgetpassword from './Pages/Forgetpassword'

const App = () => {

    // Create Function for private routing

    function PrivateRoute({ children }) {

        const token =localStorage.getItem("auth") || sessionStorage.getItem("auth");

        return token !== null && token !== undefined ? (

          children

        ) : (

          <Navigate to="/login" />

        );

      }

   // Public Route For Public Page

    const public\_route = [

        {

            path:"/login",

            component:<Login/>

        },

        {

            path:"/register",

            component:<Register/>

        },

        {

            path:"/forgetpassword",

            component:<Forgetpassword/>

        }

    ]

    // Private Route For Private Page

    const private\_route = [

        {

            path:"/",

            component:<Home/>

        }

    ]

    return (

        <>

        <ToastContainer/>

            <Router>

                <Routes>

                    {

                        public\_route?.map((value) =>{

                            return(

                                <>

                               <Route path={value.path} element={value.component}/>

                                </>

                            )

                        })

                    }

                    {

                        private\_route?.map((value) =>{

                            return(

                                <>

                                {/\*Apply Private Routing in Route\*/}

                                <Route path={value.path} element={<PrivateRoute>{value?.component}</PrivateRoute>}/>

                                </>

                            )

                        })

                    }

                </Routes>

            </Router>

        </>

    )

}

export default App

**Nav.jsx**

import React from 'react'

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

import { useAuth } from '../Context/Authcontext' // Import useAuth

import { toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

const Nav = () => {

    const [auth, setAuth] = useAuth() // Create custom hook

    // Make handle for Logout

    const handleLogout = () => {

        setAuth({

            ...auth, user: null, token: ''

        })

        localStorage.removeItem('auth')

        toast.success('Successfully logout')

    }

    return (

        <>

            <nav class="navbar navbar-expand-lg navbar-light bg-light">

                <Link class="navbar-brand" to='/'>Home</Link>

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

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

                </button>

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

                    <ul class="navbar-nav mx-auto">

                        {/\*Add condition between Login and Logout\*/}

                        {

                            !auth.user ? (

                                <>

                                    <li class="nav-item">

                                        <Link class="nav-link" to="/login">Login</Link>

                                    </li>

                                    <li class="nav-item">

                                        <Link class="nav-link" to="/register">Register </Link>

                                    </li>

                                </>

                            ) : (

                                <>

                                    <li class="nav-item">

                                        <Link class="nav-link"><b>User : </b>{auth.user.name}</Link>

                                    </li>

                                    <li class="nav-item">

                                        <Link onClick={handleLogout} class="nav-link" to="/login">Logout </Link>

                                    </li>

                                </>

                            )

                        }

                    </ul>

                </div>

            </nav>

        </>

    )

}

export default Nav

**page refresh করার পর যদি logout button হেটে যায় তো কি করবো ?**

// Make handle for Logout

const handleLogout = () => {

setAuth({

...auth, user: null, token: ''

});

localStorage.removeItem('auth');

toast.success('Successfully logout');

};

// This is what we're doing so that even after refreshing the page, the logout button doesn't disappear.

useEffect(() => {

const authData = JSON.parse(localStorage.getItem('auth'));

if (authData) {

setAuth(authData);

}

}, []);

handle Logout এর পর এই useEffect টা লাগাবো

**Important Note:** **For Authentication(private routing):Login, Register and Forget will be this process which I show. But you can do it using old step just like initialstate but there you might not be get dynamic toast error message for some cases.**

**Syntax difference try catch and then catch**

**try catch :**

try{

const response = awit fetch api/call your api function here

put positive message .......

}catch(error){

put error means negative message.......

}

**then catch :**

await fetchapi/call your api function

.then((response) =>{

put positive message ....

})

.catch((error) =>{

put negative message ....

})

then এর সময় আমরা api then এর বাইরে কল করবো কিন্তু try এর সময় api try এর ভিতর কল করবো

Always use try catch for handling API

**CRUD\_AUTHENTICATION\_TOKEN\_PROJECT**

1) Register.jsx : As same as Previous process of Form data handle no token required for Register.

2) Login.jsx : As same as Previous process of Raw data handle no token required for Login.

3) Auth.jsx : Auth.jsx and whole Authentication system are as same as previous which is present in earlier.

4) Addproduct.jsx : Addproduct.jsx with token process

import React, { useState } from 'react';

import Layout from '../Common/Layout';

import axios from 'axios';

import { useAuth } from '../Context/Authcontext'; // Import useAuth

import { toast } from 'react-toastify';

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

import Loader2 from '../Common/Loader2';

const initialstate = {

    name: '',

    price: '',

    description: '',

    brand: '',

    image: ''

};

const Addproduct = () => {

    const [addp, setAddp] = useState(initialstate);

    const [auth] = useAuth(); // Custom Hook for auth

    const [loading, setLoading] = useState(false);

    const [image, setImage] = useState('');

    const navigate = useNavigate();

    const handleOnChange = (e) => {

        const { name, value } = e.target;

        setAddp({ ...addp, [name]: value });

    };

    const handleOnSubmit = async (e) => {

        e.preventDefault();

        setLoading(true);

        const apiUrl = "https://webskitters-student.onrender.com/create/product";

        // Store token in a variable

        const mytoken = {

            headers: {

                "x-access-token": auth.token,

            },

        };

        let formdata = new FormData();

        formdata.append("name", addp.name);

        formdata.append("price", addp.price);

        formdata.append("description", addp.description);

        formdata.append("brand", addp.brand);

        formdata.append("image", image);

        try {

            const response = await axios.post(apiUrl, formdata, mytoken);

            console.log(response);

            toast.success(response?.data?.message);

            navigate('/product');

            setLoading(false);

        } catch (error) {

            console.log(error);

            toast.error(error?.response?.data?.message);

            setLoading(false);

        }

    };

    return (

        <>

            <Layout>

                {/\* <!-- Contact Start --> \*/}

                <div class="container-fluid py-5">

                    <div class="container py-5">

                        <div class="text-center mb-5">

                            <h5 class="text-primary text-uppercase mb-3" style={{ letterSpacing: '5px' }}>Product</h5>

                            <h1>Add Product</h1>

                        </div>

                        <div class="row justify-content-center">

                            <div class="col-lg-8">

                                <div class="contact-form bg-secondary rounded p-5">

                                    <div id="success"></div>

                                    <form method='post' onSubmit={handleOnSubmit}>

                                        <div class="control-group">

                                            <input type="text" class="form-control border-0 p-4" id="name" placeholder="Product Name" required="required" data-validation-required-message="Please enter product name" name='name' onChange={handleOnChange} value={addp.name} />

                                            <p class="help-block text-danger"></p>

                                        </div>

                                        <div class="control-group">

                                            <input type="number" class="form-control border-0 p-4" id="email" placeholder="Price" required="required" data-validation-required-message="Please enter price" name='price' onChange={handleOnChange} value={addp.price} />

                                            <p class="help-block text-danger"></p>

                                        </div>

                                        <div class="control-group">

                                            <textarea class="form-control border-0 py-3 px-4" rows="5" id="message" placeholder="Description" required="required" data-validation-required-message="Description" name='description' onChange={handleOnChange} value={addp.description}></textarea>

                                            <p class="help-block text-danger"></p>

                                        </div>

                                        <div class="control-group">

                                            <input type="text" class="form-control border-0 p-4" id="name" placeholder="Product brand" required="required" data-validation-required-message="Please enter product name" name='brand' onChange={handleOnChange} value={addp.brand} />

                                            <p class="help-block text-danger"></p>

                                        </div>

                                        {/\*This form section is for submit image\*/}

                                        <div className="form-group">

                                            <label for="exampleInputEmail1">Image</label>

                                            <input

                                                type="file"

                                                onChange={(e) => setImage(e.target.files[0])}

                                                name="image"

                                                accept="image/\*"

                                                class="form-control"

                                            />

                                            {image !== "" && image !== undefined && image !== null ? (

                                                <img

                                                    style={{ height: "180px" }}

                                                    // createObjectURL use for make image url

                                                    src={URL.createObjectURL(image)}

                                                    alt=""

                                                    className="upload-img"

                                                />

                                            ) : (

                                                <>{image === "" && <p>Drag or drop content here</p>}</>

                                            )}

                                        </div>

                                        {/\*Image area end\*/}

                                        <div class="text-center">

                                            <button class="btn btn-primary py-3 px-5" type="submit" id="sendMessageButton">

                                                {loading ? <Loader2/> : 'Submit'}

                                            </button>

                                        </div>

                                    </form>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

                {/\* <!-- Contact End --> \*/}

            </Layout>

        </>

    )

}

export default Addproduct

Note : const mytoken = {

            headers: {

                "x-access-token": auth.token,

            },

        };

Your token is "x-access-token" it will be change for different API

5)Allproduct.jsx: Handle get and delete API with token.

import React, { useEffect } from 'react'

import axios from 'axios'

import { useState } from 'react'

import { useAuth } from '../Context/Authcontext'

import Layout from '../Common/Layout'

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

import { toast } from 'react-toastify'

import Loader from '../Common/Loader'

const Product = () => {

    const [prod, setProd] = useState([])

    const [loading, setLoading] = useState(true)

    const [auth] = useAuth()

    const getData = async () => {

        const apiUrl = 'https://webskitters-student.onrender.com/product'

        // Store token in a variable

        const mytoken = {

            headers: {

                "x-access-token": auth.token,

            },

        };

        const response = await axios.get(apiUrl,mytoken)

        console.log("Data is fetching", response);

        setProd(response?.data?.data)

        setLoading(false)

    }

    useEffect(() => {

        getData()

    }, [])

    if (loading) {

        return (

            <h1 style={{ position: 'absolute', top: '50%', left: '50%', transform: 'translet(-50,-50)' }}><Loader /></h1>

        )

    }

    // Create a function for delete

    const deleteUserData = async (id) => {

        const deleteurl = `https://webskitters-student.onrender.com/delete/product/${id}`

        const mytoken = {

            headers: {

                "x-access-token": auth.token,

            },

        }

        try {

            const response = await axios.delete(deleteurl, mytoken);

            console.log("Response:", response);

            getData(); // Refresh data

            toast.warn(response?.data?.message);

        } catch (error) {

            console.error("Error deleting user:", error);

            toast.error(error?.response?.data?.message);

        }

    };

    return (

        <>

            <Layout>

                <div className='container' style={{ marginTop: '100px' }}>

                    <table class="table">

                        <thead>

                            <tr>

                                <th scope="col">Id</th>

                                <th scope="col">Name</th>

                                <th scope="col">Description</th>

                                <th scope="col">Image</th>

                                <th scope="col">Brand</th>

                                <th scope="col">Price</th>

                                <th scope="col">Update</th>

                                <th scope="col">Delete</th>

                            </tr>

                        </thead>

                        <tbody>

                            {prod?.slice(0, prod.length).reverse().map((value, index) => {

                                return (

                                    <>

                                        <tr>

                                            <th scope="row">{index + 1}</th>

                                            <td>{value?.name}</td>

                                            <td>{value?.description}</td>

                                            <td><img src={value?.image} alt="Product Image" style={{ height: '50px' }} /></td>

                                            <td>{value?.brand}</td>

                                            <td>{value?.price}</td>

                                            <td><Link to={`/edit/${value.\_id}`}><button type="button" class="btn btn-primary">Edit</button></Link></td>

                                            <td><button type="button" class="btn btn-danger" onClick={() => deleteUserData(value?.\_id)}>Delete</button></td>

                                        </tr>

                                    </>

                                )

                            })}

                        </tbody>

                    </table>

                </div>

            </Layout>

        </>

    )

}

export default Product

6) Editproduct.jsx : With token

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

import Layout from '../Common/Layout';

import axios from 'axios';

import { useAuth } from '../Context/Authcontext';

import { toast } from 'react-toastify';

import { useNavigate, useParams } from 'react-router-dom';

import Loader2 from '../Common/Loader2';

const initialstate = {

    name: '',

    price: '',

    description: '',

    brand: '',

    image: ''

};

const EditProducts = () => {

    const { id } = useParams();

    const [product, setProduct] = useState(initialstate);

    const [image, setImage] = useState(); // State for image

    const [loading, setLoading] = useState(false);

    const [auth] = useAuth();

    const navigate = useNavigate();

    // Function for onchange

    const handleChange = (e) => {

        const { name, value } = e.target;

        setProduct({ ...product, [name]: value })

    };

    // Function for fetch single data

    const getProduct = async () => {

        const singleAPI = `https://webskitters-student.onrender.com/edit/product/${id}`

        const myToken = {

            headers: {

                "x-access-token": auth.token,

            },

        }

        try {

            const response = await axios.get(singleAPI, myToken)

            const fitproduct = response?.data?.data;

            const reg = {

                name: fitproduct.name,

                price: fitproduct.price,

                description: fitproduct.description,

                brand: fitproduct.brand,

                image: fitproduct.image

            }

            setProduct(reg);

        } catch (error) {

            console.log(error);

        }

    };

    useEffect(() => {

        getProduct();

    }, [id, auth.token]);

    const handleSubmit = async (e) => {

        e.preventDefault();

        setLoading(true);

        const apiUrl = `https://webskitters-student.onrender.com/update/product/${id}`;

        const myToken = {

            headers: {

                "x-access-token": auth?.token,

            },

        }

        const formData = new FormData();

        formData.append("name", product.name);

        formData.append("price", product.price);

        formData.append("description", product.description);

        formData.append("brand", product.brand);

        formData.append("image", image);

        try {

            const response = await axios.post(apiUrl, formData, myToken);

            navigate("/product");

            toast.success(response?.data?.message);

            setLoading(false)

        } catch (error) {

            console.log("Error occurred:", error);

            toast.error(error?.response?.data?.message);

            setLoading(false)

        }

    };

    return (

        <>

            <Layout>

                {/\* <!-- Contact Start --> \*/}

                <div class="container-fluid py-5">

                    <div class="container py-5">

                        <div class="text-center mb-5">

                            <h5 class="text-primary text-uppercase mb-3" style={{ letterSpacing: '5px' }}>Product</h5>

                            <h1>Add Product</h1>

                        </div>

                        <div class="row justify-content-center">

                            <div class="col-lg-8">

                                <div class="contact-form bg-secondary rounded p-5">

                                    <div id="success"></div>

                                    <form method='post' onSubmit={handleSubmit}>

                                        <div class="control-group">

                                            <input type="text" class="form-control border-0 p-4" id="name" placeholder="Product Name" required="required" data-validation-required-message="Please enter product name" name='name' onChange={handleChange} value={product.name} />

                                            <p class="help-block text-danger"></p>

                                        </div>

                                        <div class="control-group">

                                            <input type="number" class="form-control border-0 p-4" id="email" placeholder="Price" required="required" data-validation-required-message="Please enter price" name='price' onChange={handleChange} value={product.price} />

                                            <p class="help-block text-danger"></p>

                                        </div>

                                        <div class="control-group">

                                            <textarea class="form-control border-0 py-3 px-4" rows="5" id="message" placeholder="Description" required="required" data-validation-required-message="Description" name='description' onChange={handleChange} value={product.description}></textarea>

                                            <p class="help-block text-danger"></p>

                                        </div>

                                        <div class="control-group">

                                            <input type="text" class="form-control border-0 p-4" id="name" placeholder="Product brand" required="required" data-validation-required-message="Please enter product name" name='brand' onChange={handleChange} value={product.brand} />

                                            <p class="help-block text-danger"></p>

                                        </div>

                                        {/\*This form section is for submit image\*/}

                                        <div className="form-group">

                                            <label for="exampleInputEmail1">photo</label>

                                            <input

                                                type="file"

                                                onChange={(e) => setImage(e.target.files[0])}

                                                name="image"

                                                accept="image/\*"

                                                class="form-control"

                                            />

                                            {image !== "" && image !== undefined && image !== null ? (

                                                <img

                                                    style={{ height: "180px" }}

                                                    // createObjectURL use for make image url

                                                    src={URL.createObjectURL(image)}

                                                    alt=""

                                                    className="upload-img"

                                                />

                                            ) : (

                                                <>{image === "" && <p>Drag or drop content here</p>}</>

                                            )}

                                        </div>

                                        {/\*Image area end\*/}

                                        <div class="text-center">

                                            <button class="btn btn-primary py-3 px-5" type="submit" id="sendMessageButton">

                                                {loading ? <Loader2 /> : 'Edit'}

                                            </button>

                                        </div>

                                    </form>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

                {/\* <!-- Contact End --> \*/}

            </Layout>

        </>

    )

}

export default EditProducts

Note : You will do always Editproduct in this process when Token is required or non required.

7) Updatepassword.jsx : With token

import React, { useState } from 'react'

import axios from 'axios'

import { toast } from 'react-toastify';

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

import { useAuth } from '../Context/Authcontext';

import Layout from '../Common/Layout';

import Loader2 from '../Common/Loader2';

const initialstate = {

    user\_id: '',

    password: ''

}

const Updatepassword = () => {

    const [update, setUpdate] = useState(initialstate);

    const [loading, setLoading] = useState(false);

    const [auth] = useAuth();

    const navigate = useNavigate();

    const handleOnChange = (e) => {

        const { name, value } = e.target

        setUpdate({ ...update, [name]: value })

    }

    const handleOnSubmit = async (e) => {

        e.preventDefault()

        setLoading(true)

        const updateUrl = "https://webskitters-student.onrender.com/update-password"

        const mytoken = {

            headers: {

                "x-access-token": auth.token,

            },

        };

        try {

            const response = await axios.post(updateUrl, update, mytoken)

            if (response && response?.data?.success === true) {

                console.log(response);

                toast.success(response?.data?.msg)

                setLoading(false)

                navigate('/dashboard')

            } else {

                console.log(response);

                toast.error(response?.data?.msg)

            }

        } catch (error) {

            console.log("Error doing update", error);

            setLoading(false)

        }

    }

    return (

        <>

            <Layout>

                {/\* <!-- Login Start --> \*/}

                <div class="container-fluid bg-registration py-5" style={{ margin: '90px 0' }}>

                    <div class="container py-5">

                        <div class="row align-items-center">

                            <div class="col-md-6 mx-auto">

                                <div class="card border-0">

                                    <div class="card-header bg-light text-center p-4">

                                        <h1 class="m-0">Update Password</h1>

                                    </div>

                                    <div class="card-body rounded-bottom bg-primary p-5">

                                        <form method='post' onSubmit={handleOnSubmit}>

                                            <div class="form-group">

                                                <input type="text" class="form-control border-0 p-4" placeholder="User ID" required="required" name='user\_id' value={update.user\_id} onChange={handleOnChange} />

                                            </div>

                                            <div class="form-group">

                                                <input type="password" class="form-control border-0 p-4" placeholder="Your password" required="required" name='password' value={update.password} onChange={handleOnChange} />

                                            </div>

                                            <div>

                                                <button class="btn btn-dark btn-block border-0 py-3" type="submit">

                                                    {loading ? <Loader2/> : 'Update'}

                                                </button>

                                            </div>

                                        </form>

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

                {/\* <!-- Login End --> \*/}

            </Layout>

        </>

    )

}

export default Updatepassword

8)Forgetpassword.jsx: With token

import axios from 'axios';

import React, { useState } from 'react';

import { toast } from 'react-toastify';

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

import Layout from '../Common/Layout';

import { useAuth } from '../Context/Authcontext';

import Loader2 from '../Common/Loader2';

const initialState = {

    email: '',

    first\_school: '',

    newPassword: ''

};

const Forget = () => {

    const [forget, setForget] = useState(initialState);

    const [loading, setLoading] = useState(false);

    const navigate = useNavigate();

    const [auth] = useAuth();

    const handleOnChange = (e) => {

        const { name, value } = e.target;

        setForget({ ...forget, [name]: value });

    };

    const handleOnSubmit = async (e) => {

        e.preventDefault();

        setLoading(true);

        const forgetUrl = "https://webskitters-student.onrender.com/forget-password";

        const myToken = {

            headers: {

                "x-access-token": auth.token,

            },

        };

        try {

            const response = await axios.post(forgetUrl, forget, myToken);

            if (response && response.data && response.data.success) {

                toast.success(response.data.message);

                setLoading(false);

                navigate('/login');

            } else {

                toast.error(response.data.message);

                setLoading(false);

            }

        } catch (error) {

            console.error("Error forgotten password", error);

            toast.error(error.response ? error.response.data.message : "An error occurred");

            setLoading(false);

        }

    };

    return (

        <Layout>

            <div className="container-fluid bg-registration py-5" style={{ margin: '90px 0' }}>

                <div className="container py-5">

                    <div className="row align-items-center">

                        <div className="col-md-6 mx-auto">

                            <div className="card border-0">

                                <div className="card-header bg-light text-center p-4">

                                    <h1 className="m-0">Forget Password</h1>

                                </div>

                                <div className="card-body rounded-bottom bg-primary p-5">

                                    <form onSubmit={handleOnSubmit}>

                                        <div className="form-group">

                                            <input type="email" className="form-control border-0 p-4" placeholder="Your email" required name='email' value={forget.email} onChange={handleOnChange} />

                                        </div>

                                        <div className="form-group">

                                            <input type="text" className="form-control border-0 p-4" placeholder="First School" required name='first\_school' value={forget.first\_school} onChange={handleOnChange} />

                                        </div>

                                        <div className="form-group">

                                            <input type="password" className="form-control border-0 p-4" placeholder="New password" required name='newPassword' value={forget.newPassword} onChange={handleOnChange} />

                                        </div>

                                        <div>

                                            <button className="btn btn-dark btn-block border-0 py-3" type="submit">

                                                {loading ? <Loader2/> : 'Forget'}

                                            </button>

                                        </div>

                                    </form>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </Layout>

    );

};

export default Forget;

8) Dashboard.jsx : Using Auth

import React from 'react'

import Layout from '../Common/Layout'

import { useAuth } from '../Context/Authcontext' // Import useAuth

const Dashboard = () => {

    const [auth] = useAuth(); // Custom Hook

    return (

        <>

            <Layout>

                {/\* <!--About section with Grid start--> \*/}

                <div class="container">

                    <div class="row">

                        <div class="col text-center mt-5">

                            <h1>User</h1>

                        </div>

                    </div>

                    <div class="row mt-5">

                        <div class="col-md-6 align-self-center">

                            <img src={auth?.user?.image} alt="Error" style={{ maxWidth: '100%' }} />

                        </div>

                        <div class="col-md-6 p-3  align-self-center">

                            <li style={{ fontSize: '30px', listStyle: 'none' }}><b>Id : </b>{auth?.user?.\_id}</li>

                            <li style={{ fontSize: '30px', listStyle: 'none' }}><b>Name : </b>{auth?.user?.name}</li>

                            <li style={{ fontSize: '30px', listStyle: 'none' }}><b>Email : </b>{auth?.user?.email}</li>

                            <li style={{ fontSize: '30px', listStyle: 'none' }}><b>First School : </b>{auth?.user?.first\_school}</li>

                            <li style={{ fontSize: '30px', listStyle: 'none' }}><b>Role : </b>{auth?.user?.role}</li>

                        </div>

                    </div>

                </div>

                {/\* <!--About section with Grid end--> \*/}

            </Layout>

        </>

    )

}

export default Dashboard

For to Get Full project: <https://github.com/mousomray/crud_auth_token_project>

**Done Private Routing, Authentication and Token**

42) What is useMemo and useCallback ?

The useMemo and useCallback Hooks are similar. The main difference is that useMemo returns a memoized value and useCallback returns a memoized function. You can learn more about useCallback in the [useCallback chapter](https://www.w3schools.com/react/react_usecallback.asp).

Memochild.jsx

import React, { memo } from 'react'

const Child = (name) => {

console.log("I am child",name);

  return (

    <>

      <h1>I am Child</h1>

    </>

  )

}

export default memo(Child)

Memoparent.jsx

import React, { useState } from 'react'

import Child from './Memochild'

const Parent = () => {

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

const increase = () =>{

    addCount(count + 1);

}

  return (

    <>

      <h1>I am Parent</h1>

      <Child name="pritam"/>

      <h1>{count}</h1>

      <button onClick={increase}>Increase</button>

    </>

  )

}

export default Parent

**43)What is React Hook Form?**

React Hook Form is a library for managing form state and validation in React applications using hooks.

What is register in React Hook Form?

In React Hook Form, "register" is a function used to register input components and connect them to the form's state.

What is watch in React Hook Form?

In React Hook Form, "watch" is a function used to track changes to specific form fields and retrieve their current values.

How to do it in Practical ?

Step1: npm install react-hook-form

Step2: Main coding Part

import React, { useState } from 'react'

import axios from 'axios'

import { ToastContainer, toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

import { useForm } from 'react-hook-form'; // Import useForm hook

const Adddata = () => {

    const { register, handleSubmit, watch, formState: { errors } } = useForm(); // Define in State

    const[loading, setLoading] = useState(false)

    const handleOnSubmit = async (data) => {

        setLoading(true)

        const reg = {

            name: data.name,

            email: data.email,

            phone: data.phone,

            city: data.city,

        };

        const apiUrl = "https://tureappservar.onrender.com/user"

        try {

            const response = await axios.post(apiUrl, reg, data)

            console.log(response);

            toast.success(response?.data?.message)

            setLoading(false)

        } catch (error) {

            console.log(error);

            toast.error(error?.response?.data?.message)

            setLoading(false)

        }

    }

    return (

        <>

            <ToastContainer />

            <div classNameName="container">

                <div className="card">

                    <div className="card-header">

                        SignUp Form React Hook Form

                    </div>

                    <div className="card-body">

                        <form onSubmit={handleSubmit(handleOnSubmit)}>

                            <div className="form-group">

                                <label for="exampleInputEmail1">Name</label>

                                <input type="text" className="form-control"

                                    {...register("name", {

                                        required: true,

                                        minLength:3

                                    })}

                                />

                                {errors?.name?.type === "required" && <p style={{ color: 'red' }}>This field is required</p>}

                                {errors?.name?.type === "minLength" && <p style={{ color: 'red' }}>Name must be atleast three characters</p>}

                            </div>

                            <div className="form-group">

                                <label htmlFor="exampleInputEmail1">Email address</label>

                                <input

                                    type="email"

                                    className="form-control"

                                    {...register("email", {

                                        required: "This field is required",

                                        pattern: {

                                            value: /^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)\*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/,

                                            message: "Email Pattern should be xyz@gmail.com",

                                        },

                                    })}

                                />

                                {errors.email && <p style={{ color: 'red' }}>{errors.email.message}</p>}

                            </div>

                            <div className="form-group">

                                <label htmlFor="phone">Phone</label>

                                <input

                                    type="number"

                                    className="form-control"

                                    {...register("phone", {

                                        required: true,

                                        minLength: 10,

                                        maxLength: 10,

                                    })}

                                />

                                {errors?.phone?.type === "required" && (

                                    <p style={{ color: 'red' }}>This field is required</p>

                                )}

                                {errors?.phone?.type === "minLength" && (

                                    <p style={{ color: 'red' }}>Phone number must be exactly 10 characters</p>

                                )}

                                {errors?.phone?.type === "maxLength" && (

                                    <p style={{ color: 'red' }}>Phone number must be exactly 10 characters</p>

                                )}

                            </div>

                            <div className="form-group">

                                <label for="exampleInputEmail1">city</label>

                                <input type="text" className="form-control"

                                    {...register("city", {

                                        required: true,

                                    })}

                                />

                                {errors?.city?.type === "required" && <p style={{ color: 'red' }}>This field is required</p>}

                            </div>

                            {/\* <div className="form-group">

                                <label for="exampleInputPassword1">Password</label>

                                <input type="password" className="form-control"

                                    name='password'

                                    {...register("password", {

                                        required: true,

                                        maxLength: 10,

                                    })}

                                />

                                {errors?.password?.type === "required" && <p>This field is required</p>}

                                {errors?.password?.type === "maxLength" && (

                                    <p>First phone cannot exceed 10 characters</p>

                                )}

                            </div> \*/}

                            <button type="submit" className="btn btn-primary">

                                {loading?'Loading...':'Submit'}

                            </button>

                        </form>

                    </div>

                </div>

            </div>

        </>

    )

}

export default Adddata

Note: const { register, handleSubmit, watch, formState: { errors } } = useForm(); // Define in State

You can’t change any name in this code

We will use this process for validation it is the substitute process of handleOnChange

**44) What is React Query**

React Query is a library for managing, caching, and synchronizing server state in React applications.

We use React Query to simplify state management, caching, and server data synchronization in React applications.

How to do it in Practical ?

Step1: Go to VS code terminal > npm i @tanstack/react-query you can copy paste this code from <https://tanstack.com/query/latest/docs/framework/react/installation> this site.

**Step2:**

import {

    QueryClient,

    QueryClientProvider,

} from '@tanstack/react-query'

put this code in your App.js file but in this case my parent component I make Routequery.jsx

**Step3:** তোমার রাউটিং এর পার্টকে <QueryclientProvider> দিয়ে মুড়ে দাও এবং আরো কিছু জিনিস Add করো।

Just like this:

Routequery.jsx : My parent component

import React from 'react'

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

import Nav from './Nav'

import User from './User'

// Import QueryClient for React Query

import { QueryClient, QueryClientProvider } from '@tanstack/react-query'

const Routequery = () => {

    // Create a client

    const queryClient = new QueryClient()

    return (

        <>

            {/\*Cover with QueryClientProvider\*/}

            <QueryClientProvider client={queryClient}>

                <Router>

                    <Nav />

                    <Routes>

                        <Route path='/' element={<User />} />

                    </Routes>

                </Router>

            </QueryClientProvider>

        </>

    )

}

export default Routequery

User.jsx:

import React from 'react'

import { useQuery } from '@tanstack/react-query' // Import for useQuery

import axios from 'axios'

const User = () => {

    const getUserdata = async () => {

        const userapiurl = 'https://tureappservar.onrender.com/alluser'

        const response = await axios.get(userapiurl)

        console.log("Fetching User data", response);

        return response?.data?.data // You have to put return because there is no state

    }

    const { isLoading, isError, data, error } = useQuery({

        queryKey: ['user'], // You can write any name in the place of user

        queryFn: getUserdata // This line of code work as same as useEffect()

    })

    // For Loading

    if (isLoading) {

        return <h1>Loading...</h1>

    }

    // For Error

    if (isError) {

        return <h1>{error.message}</h1>

    }

    return (

        <>

            <h1>User Page</h1>

            {

                data?.map((value) => {

                    return (

                        <>

                            <div key={value.\_id}>

                                <h1>{value?.name}</h1>

                            </div>

                        </>

                    )

                })

            }

        </>

    )

}

export default User

Note:It is a get data so I use useQuery(). For post data we have to use useMutation()

Register.jsx (Here i applied useMutation)

import React, { useState } from "react";

import Avatar from "@mui/material/Avatar";

import Button from "@mui/material/Button";

import CssBaseline from "@mui/material/CssBaseline";

import TextField from "@mui/material/TextField";

import Link from "@mui/material/Link";

import Grid from "@mui/material/Grid";

import Box from "@mui/material/Box";

import LockOutlinedIcon from "@mui/icons-material/LockOutlined";

import Typography from "@mui/material/Typography";

import Container from "@mui/material/Container";

import Paper from "@mui/material/Paper";

import { createTheme, ThemeProvider } from "@mui/material/styles";

import { useForm } from "react-hook-form";

import { useMutation } from "@tanstack/react-query";

import axios from "axios";

const defaultTheme = createTheme();

const Register\_URL = "https://restapinodejs.onrender.com/api/register";

const Register = () => {

// For Image Handling

const [selectedImage, setSelectedImage] = useState(null);

const [img, setImg] = useState(null);

const {

register,

handleSubmit,

formState: { errors },

reset,

} = useForm();

const registerUser = async (data, img) => {

// Handling Form Data Area

const formData = new FormData();

formData.append("name", data.name);

formData.append("email", data.email);

formData.append("mobile", data.mobile);

formData.append("password", data.password);

formData.append("photo", img);

const response = await axios.post(Register\_URL, formData);

return response.data;

};

const mutation = useMutation({

mutationFn: (data) => registerUser(data, img),

onSuccess: (data) => {

console.log("Data submitted successfully:", data);

reset(); // Blank form after submitting data

//navigate("/login");

},

onError: (error) => {

console.error("Error submitting data:", error);

// Add your error handling logic here

},

});

const onSubmit = (data) => {

mutation.mutate(data);

};

const handleImageChange = (event) => {

const file = event.target.files[0];

setImg(file);

if (file) {

setSelectedImage(URL.createObjectURL(file));

}

};

return (

<ThemeProvider theme={defaultTheme}>

<Container component="main" maxWidth="xs">

<Paper

elevation={5}

style={{

padding: "1rem 3rem",

marginTop: "1rem",

width: "35rem",

marginBottom: "1rem",

}}

>

<CssBaseline />

<Box

sx={{

marginTop: 8,

display: "flex",

flexDirection: "column",

alignItems: "center",

}}

>

<Avatar sx={{ m: 1, bgcolor: "secondary.main" }}>

<LockOutlinedIcon />

</Avatar>

<Typography component="h1" variant="h5">

Sign up

</Typography>

<Box

component="form"

noValidate

onSubmit={handleSubmit(onSubmit)}

sx={{ mt: 3 }}

>

<Grid container spacing={2}>

<Grid item xs={12}>

<TextField

required

fullWidth

id="name"

label="Name"

{...register("name", { required: true, maxLength: 20 })}

/>

{errors.name && (

<p style={{ color: "red" }}>This field is required</p>

)}

</Grid>

<Grid item xs={12}>

<TextField

required

fullWidth

id="email"

label="Email Address"

{...register("email", { required: true, maxLength: 50 })}

autoComplete="email"

/>

{errors.email && (

<p style={{ color: "red" }}>This field is required</p>

)}

</Grid>

<Grid item xs={12}>

<TextField

required

fullWidth

id="mobile"

label="Mobile"

{...register("mobile", { required: true, maxLength: 20 })}

/>

{errors.mobile && (

<p style={{ color: "red" }}>This field is required</p>

)}

</Grid>

<Grid item xs={12}>

<TextField

required

fullWidth

{...register("password", { required: true, maxLength: 20 })}

label="Password"

type="password"

id="password"

autoComplete="new-password"

/>

{errors.password && (

<p style={{ color: "red" }}>This field is required</p>

)}

</Grid>

<Grid item xs={12}>

<TextField

fullWidth

type="file"

accept="image/\*"

onChange={handleImageChange}

id="photo"

/>

</Grid>

{selectedImage && (

<Grid item xs={12}>

<img

src={selectedImage}

alt="Selected Image"

style={{ maxWidth: "100%" }}

/>

</Grid>

)}

</Grid>

<Button

type="submit"

fullWidth

variant="contained"

sx={{ mt: 3, mb: 2 }}

disabled={mutation.isLoading}

>

{mutation.isLoading ? "Loading..." : "Sign Up"}

</Button>

<Grid container justifyContent="flex-end">

<Grid item>

<Link href="#" variant="body2">

Already have an account? Sign in

</Link>

</Grid>

</Grid>

</Box>

</Box>

</Paper>

</Container>

</ThemeProvider>

);

};

export default Register;

45)CRUD এ Read এর সময় যদি Post data api দেয় তাহলে কি করবো ?

Ans: const response = await axios.post(apiurl, piller, mytoken)

শুধুমাত্র এই লাইনটাতেই change হবে get এর জায়গায় post করবো আর স্টেট ভ্যারিয়েবল টাকে ব্রাকেটের মধ্যে রাখব। আর বাদ বাকি As Same As Previous process.

const [piller, changable] useState([]) in this code what is piller and what is changable ?

piller is a state variable, and changable is a function to update its value, created using the useState hook in React.

46)CRUD এ Delete এর সময় যদি Post data api দেয় তাহলে কি করবো ?

Ans: const response = await axios.post(apiurl, { id }, mytoken);

আমরা যেখানে backtick দিয়ে api url এ Id পাস করতাম সেটা না করে id টাকে As a object এই ভাবে পাস করব। আর বাদ বাকি As Same As Previous process.

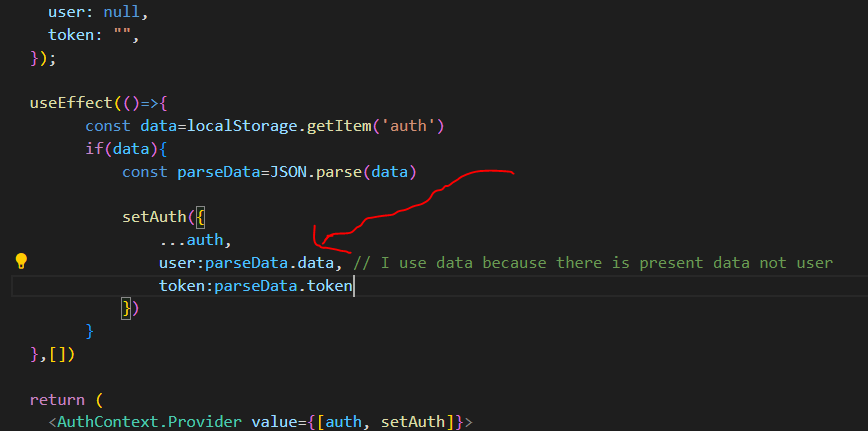
46)CRUD এ Edit এর সময় যদি Id is Required বলে দেয় তাহলে কি করবো ?

Ans: formData.append("id", id) // In this API, Id is required so you have to put id in formdata. আর বাদ বাকি As Same As Previous process.

46)Authentication সময় একটা জিনিস মাথায় রাখার বিষয় :

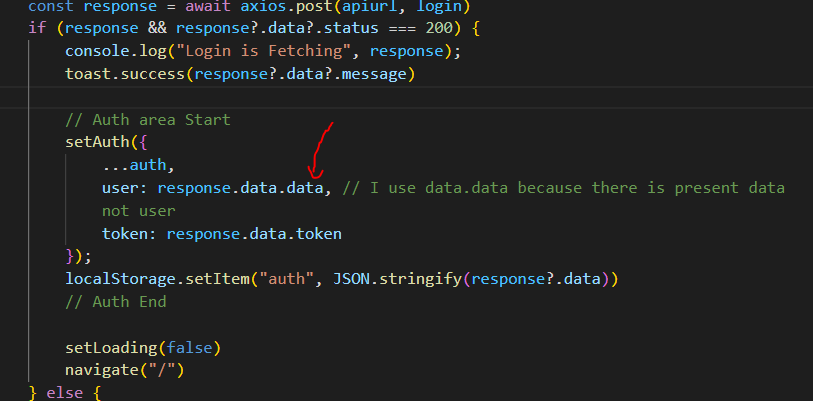
Ans: যদি login ডাটাতে user না দিয়ে data দেয় বা অন্য কিছু দেয় তাহলে

In Auth.jsx



এখানে user এর জায়গায় data বা যা থাকবে তা লেখো যেমনটি এখানে লেখা আছে

Login.jsx



এখানে user এর জায়গায় data বা যা থাকবে তা লেখো যেমনটি এখানে লেখা আছে

**47) How to Filter Data by category ?**

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

import axios from 'axios';

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

const Product = () => {

    const [products, setProducts] = useState([]); // For to fetch Products

    const [categories, setCategories] = useState([]); // For to fetch categories

    const [loading, setLoading] = useState(true); // For to make loading

    const [visibleCards, setVisibleCards] = useState(6); // For Loadmore button

    const [selectedCategory, setSelectedCategory] = useState(''); // For selected category

    const fetchData = async () => {

        // Fetch Product area

        try {

            const apiurl = 'https://dummyjson.com/products?skip=0&limit=100'

            const response = await axios.get(apiurl);

            setProducts(response.data.products);

            setLoading(false);

        } catch (error) {

            console.error('Error fetching products data:', error);

            setLoading(false);

        }

        // Fetch Category area

        try {

            const apiurl = 'https://dummyjson.com/products/categories'

            const response = await axios.get(apiurl);

            setCategories(response.data);

        } catch (error) {

            console.error('Error fetching categories data:', error);

        }

    };

    useEffect(() => {

        fetchData();

    }, []);

    const handleLoadMore = () => {

        setVisibleCards(prevVisibleCards => prevVisibleCards + 6);

    };

    // Handle for category click

    const handleCategoryClick = (categoryId) => {

        setSelectedCategory(categoryId);

    };

    // Handle for all category

    const handleShowAll = () => {

        setSelectedCategory('');

    }

    // Filter Product

    const filteredProducts = selectedCategory

        ? products.filter(item => item.category === selectedCategory)

        : products;

    if (loading) {

        return <h1 style={{ fontSize: '100px', color: 'red', display: 'flex', justifyContent: 'center', alignItems: 'center', height: '100vh' }}>Loading....</h1>;

    }

    return (

        <div className="container mt-5">

            <div className="row">

                <div className="col-md-3 mt-5" style={{ border: '2px solid black', borderRadius: '30px', height: '1100px' }}>

                    <div>

                        <h1 style={{ marginBottom: '30px' }}>Categories</h1>

                        <ul>

                            <li style={{ marginBottom: '10px' }}>

                                <button className="btn btn-danger" onClick={handleShowAll}>

                                    <b>All</b>

                                </button>

                            </li>

                            {categories.map((value,index) => (

                                <li key={index} style={{ marginBottom: '10px' }}>

                                    <button className="btn btn-danger" onClick={() => handleCategoryClick(value)}>

                                    {/\*In this API id is not present so I am not put value.id in handleCategory Click\*/}

                                        <b>{value}</b>

                                    </button>

                                </li>

                            ))}

                        </ul>

                    </div>

                </div>

                <div className="col-md-9">

                    <div className="row">

                        {filteredProducts.slice(0, visibleCards).map(product => (

                            <div className="col-md-4" key={product.id}>

                                <div className="card mb-5 mt-5" style={{ height: '600px' }}>

                                    <img src={product.thumbnail} className="card-img-top" alt="..." style={{ height: '300px' }} />

                                    <div className="card-body">

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

                                        <p className="card-text">{product.description}</p>

                                        <Link to={`/productdetails/${product.id}`} className="btn btn-primary">Details</Link>

                                    </div>

                                </div>

                            </div>

                        ))}

                    </div>

                </div>

            </div>

            {visibleCards < filteredProducts.length && (

                <div className="text-center mt-3">

                    <button className="btn btn-success" onClick={handleLoadMore}>Load More</button>

                </div>

            )}

        </div>

    );

};

export default Product;

48) How to Post and Show Like, Unlike and Comment in any blog ? (This code area is the part of blog\_course\_project all API are given)

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

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

import Layout from '../Common/Layout';

import Loader2 from '../Common/Loader2';

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

import axios from 'axios';

import { useAuth } from '../Context/AuthContext';

import { toast } from 'react-toastify';

import Loader1 from '../Common/Loader1';

import ThumbUpIcon from '@mui/icons-material/ThumbUp'; // MUI Thumb Up Icon

import ThumbDownIcon from '@mui/icons-material/ThumbDown'; // MUI Thumb Down Icon

import CommentIcon from '@mui/icons-material/Comment'; // For Comment Icon

const initialstate = {

    name: '',

    email: '',

    comment: ''

};

const Blogdetails = () => {

    const [commentpost, setcommentPost] = useState(initialstate); // For Comment Post

    const [comment, setComment] = useState([]); // For Show Comment normal visibility

    const [showComments, setShowComments] = useState(false); // State to control comment visibility after click comment Icon

    const [like, setLike] = useState([]); // For Like count

    const [unlike, setunLike] = useState([]); // For Unlike count

    const [blogdetails, setblogDetails] = useState([]); // For Blog Details

    const [loadmore, setLoadmore] = useState(3); // For Loadmore

    const [loading, setLoading] = useState(true); // For Loading1

    const [loader, setLoader] = useState(false); // For Loading2

    const [auth] = useAuth(); // Custom Hook

    const { id } = useParams(); // Use Params

    // Like, Comment and Blog Get Handling Start

    // For Blog Details

    const getData = async () => {

        const apiurl = `https://restapinodejs.onrender.com/api/blogdetails/${id}`;

        // Store token in a variable

        const mytoken = {

            headers: {

                "x-access-token": auth.token,

            },

        };

        const response = await axios.get(apiurl, mytoken);

        console.log(response);

        setblogDetails(response?.data?.data);

        setLoading(false);

    };

    // For to show Comment

    const getComment = async () => {

        try {

            const apiurl = `https://restapinodejs.onrender.com/api/comment/${id}`;

            // Store token in a variable

            const mytoken = {

                headers: {

                    "x-access-token": auth.token,

                },

            };

            const response = await axios.get(apiurl, mytoken);

            console.log("Comment data is fetching", response);

            setComment(response?.data?.post?.comment?.comments);

        } catch (error) {

            console.log("Error Fetching Data", error);

            throw error;

        }

    };

    // Like Data Fetching for to Show Like

    const getlike = async () => {

        try {

            const apiurl = `https://restapinodejs.onrender.com/api/blog/like/${id}`;

            // Store token in a variable

            const mytoken = {

                headers: {

                    "x-access-token": auth.token,

                },

            };

            const response = await axios.put(apiurl, {}, mytoken);

            console.log("Like Data is Fetching", response);

            setLike(response?.data);

        } catch (error) {

            console.log("Error Fetching Like Data", error);

        }

    };

    // Unlike Data Fetching for to Show Like

    const getunlike = async () => {

        try {

            const apiurl = `https://restapinodejs.onrender.com/api/blog/unlike/${id}`;

            // Store token in a variable

            const mytoken = {

                headers: {

                    "x-access-token": auth.token,

                },

            };

            const response = await axios.put(apiurl, {}, mytoken);

            console.log("Unlike Data is Fetching", response);

            setunLike(response?.data);

        } catch (error) {

            console.log("Error Fetching Unlike Data", error);

        }

    };

    useEffect(() => {

        getData(); // For Blog

        getComment(); // For Comment Show

        getlike(); // For to Show Like

        getunlike(); // For to Show Unlike

    }, []);

    // For Loadmore

    const handleLoadmore = () => {

        setLoadmore(prev => prev + 4);

    };

    if (loading) {

        return <h1 style={{ position: 'absolute', top: '50%', left: '50%', transform: 'translate(-50%, -50%)' }}><Loader1 /></h1>;

    }

    // Comment Get Handling Done

    // Comment Post Handling

    const handleOnChange = (e) => {

        const { name, value } = e.target;

        setcommentPost({ ...commentpost, [name]: value });

    };

    const handleOnSubmit = async (e) => {

        e.preventDefault();

        setLoader(true);

        try {

            const apiurl = `https://restapinodejs.onrender.com/api/blog/${id}/comment/create`;

            // Store token in a variable

            const mytoken = {

                headers: {

                    "x-access-token": auth.token,

                },

            };

            const response = await axios.post(apiurl, commentpost, mytoken);

            console.log("Comment posting API is fetching", response);

            toast.success(response?.data?.message);

            setcommentPost(initialstate);

            setLoader(false);

            getComment(); // Show Comment After Submition so call getComment function

        } catch (error) {

            console.log(error);

            toast.error(error?.response?.data?.message);

            setLoader(false);

        }

    };

    // Increase like

    const handleLike = async () => {

        try {

            const apiurl = `https://restapinodejs.onrender.com/api/blog/like/${id}`;

            // Store token in a variable

            const mytoken = {

                headers: {

                    "x-access-token": auth.token,

                },

            };

            const response = await axios.put(apiurl, {}, mytoken);

            console.log("Like increased", response);

            setLike(response?.data);

        } catch (error) {

            console.log("Error increasing like", error);

        }

    };

    // Increase unlike

    const handleUnlike = async () => {

        try {

            const apiurl = `https://restapinodejs.onrender.com/api/blog/unlike/${id}`;

            // Store token in a variable

            const mytoken = {

                headers: {

                    "x-access-token": auth.token,

                },

            };

            const response = await axios.put(apiurl, {}, mytoken);

            console.log("Unlike increased", response);

            setunLike(response?.data);

        } catch (error) {

            console.log("Error increasing unlike", error);

        }

    };

    return (

        <>

            <Layout>

                {/\* <!-- ======= Blog Section ======= --> \*/}

                <section id="blog" className="blog" style={{ marginTop: '130px' }}>

                    <div className="container">

                        <div className="row">

                            <div className="col-lg-12 entries">

                                <article className="entry entry-single" data-aos="fade-up">

                                    <div className="entry-img">

                                        <img src={`https://restapinodejs.onrender.com/api/blog/image/${id}`} alt="" />

                                    </div>

                                    <h2 className="entry-title">

                                        <a href="blog-single.html">{blogdetails?.title}</a>

                                    </h2>

                                    <div className="entry-content">

                                        <p

                                            dangerouslySetInnerHTML={{ \_\_html: blogdetails?.postText }}

                                        />

                                    </div>

                                    <div className='entry-content'>

                                        {/\*Create date time area\*/}

                                        <p className='text-muted'>Created date : {new Date(blogdetails.createdAt).toLocaleDateString('en-GB')}</p>

                                        <p className='text-muted'>Created time: {new Date(blogdetails.createdAt).toLocaleTimeString()}</p>

                                        {/\*Update date time area\*/}

                                        <p className='text-muted'>Updated date : {new Date(blogdetails.updatedAt).toLocaleDateString('en-GB')}</p>

                                        <p className='text-muted'>Updated time: {new Date(blogdetails.updatedAt).toLocaleTimeString()}</p>

                                    </div>

                                    <div className="entry-footer clearfix">

                                        <div className="float-left">

                                            <i className="icofont-folder"></i>

                                            <ul className="cats">

                                                <li><a href="#">Business</a></li>

                                            </ul>

                                            <i className="icofont-tags"></i>

                                            <ul className="tags">

                                                <li><a href="#">Creative</a></li>

                                                <li><a href="#">Tips</a></li>

                                                <li><a href="#">Marketing</a></li>

                                            </ul>

                                        </div>

                                        <div className="float-right share">

                                            <a href="" title="Share on Twitter"><i className="icofont-twitter"></i></a>

                                            <a href="" title="Share on Facebook"><i className="icofont-facebook"></i></a>

                                            <a href="" title="Share on Instagram"><i className="icofont-instagram"></i></a>

                                        </div>

                                    </div>

                                </article>

                                {/\* <!-- End blog entry --> \*/}

                                {/\*Start Like, Unlike and Comment Area \*/}

                                <div className="comment-area">

                                    <ThumbUpIcon onClick={handleLike} style={{ cursor: 'pointer' }} /> {like?.likes} &nbsp;&nbsp;&nbsp;

                                    <ThumbDownIcon onClick={handleUnlike} style={{ cursor: 'pointer' }} /> {unlike?.unlikes} &nbsp;&nbsp;&nbsp;

                                    <CommentIcon onClick={() => setShowComments(!showComments)} style={{ cursor: 'pointer' }} /> {comment.length} Comments

                                    {showComments && (

                                        <div className="blog-comments mt-5" data-aos="fade-up">

                                            {comment?.slice(0, comment?.length).reverse().slice(0, loadmore).map((value) => {

                                                return (

                                                    <div id="comment" className="comment clearfix" style={{ border: '2px solid #ccc', padding: '10px', marginBottom: '20px' }}>

                                                        <h5><b>Name : </b>{value?.name}</h5>

                                                        <p><b>Email : </b>{value?.email}</p>

                                                        <p><b>Comment : </b>{value?.comment}</p>

                                                        {/\*Create date time area\*/}

                                                        <p className='text-muted'>Created date : {new Date(value.createdAt).toLocaleDateString('en-GB')}</p>

                                                        <p className='text-muted'>Created time: {new Date(value.createdAt).toLocaleTimeString()}</p>

                                                        {/\*Update date time area\*/}

                                                        <p className='text-muted'>Updated date : {new Date(value.updatedAt).toLocaleDateString('en-GB')}</p>

                                                        <p className='text-muted'>Updated time: {new Date(value.updatedAt).toLocaleTimeString()}</p>

                                                    </div>

                                                )

                                            })}

                                            {loadmore < comment.length ? (

                                                <div className="text-center mt-3">

                                                    <Link><p onClick={handleLoadmore} style={{ height: '50px' }}>See More</p></Link>

                                                </div>

                                            ) : null}

                                            <div className="reply-form">

                                                <h4>Leave a Reply</h4>

                                                <p>Your email address will not be published. Required fields are marked \* </p>

                                                <form method='post' onSubmit={handleOnSubmit}>

                                                    <div className="row">

                                                        <div className="col-md-6 form-group">

                                                            <input name="name" type="text" className="form-control" placeholder="Your Name\*" value={commentpost?.name} onChange={handleOnChange} />

                                                        </div>

                                                        <div className="col-md-6 form-group">

                                                            <input name="email" type="text" className="form-control" placeholder="Your Email\*" value={commentpost?.email} onChange={handleOnChange} />

                                                        </div>

                                                    </div>

                                                    <div className="row">

                                                        <div className="col form-group">

                                                            <textarea name="comment" className="form-control" placeholder="Your Comment\*" value={commentpost?.comment} onChange={handleOnChange}></textarea>

                                                        </div>

                                                    </div>

                                                    <button type="submit" className="btn btn-primary">

                                                        {loader ? <Loader2 /> : 'Post Comment'}

                                                    </button>

                                                </form>

                                            </div>

                                        </div>

                                    )}

                                </div>

                                {/\* <!-- End blog comments --> \*/}

                            </div>

                            {/\* <!-- End blog entries list --> \*/}

                        </div>

                    </div>

                </section>

                {/\* <!-- End Blog Section --> \*/}

            </Layout >

        </>

    )

}

export default Blogdetails;

49) How to show date time in Perfect Pattern ?

 {/\*Date area\*/}

                                                        <p className='text-muted'>Created date : {new Date(value.createdAt).toLocaleDateString('en-GB')}</p>

{/\*Time area\*/}

                                                        <p className='text-muted'>Created time: {new Date(value.createdAt).toLocaleTimeString()}</p>

Note: For Date you put toLocalDateString() and For time you put toLocalTimeString() (‘en-GB’) is use for English pattern time. (value.createdAt) is my dynamic data.

**Redux**

**50) What is Redux ? And why we use it ?**

Redux is a predictable state container for JavaScript apps, commonly used with React for managing application state in a centralized manner.

What is useSelector in Redux ?

useSelector is a hook provided by React Redux that allows components to extract data from the Redux store state.

What is configureStore in Redux ?

configureStore is a function from Redux Toolkit that simplifies the setup of Redux store by combining reducers, middleware, and dev tools configuration.

What is Redux Store ?

Redux Store is a JavaScript object that holds the entire state tree of your application, allowing access to state and dispatching actions.

What is Dispatching in Redux ?

Dispatching in Redux is the process of sending an action to the Redux store to update the state.

What is Provider in Redux ?

Provider is a component provided by React Redux that allows Redux to inject the store into the React component hierarchy, enabling access to the Redux store from any component.

what is createReducer inReact ?

`createReducer` is a function in React used to manage state changes in Redux by handling actions and returning new state immutably.

Why we use builder in Redux ?

Builders in Redux are used to create action objects with predefined structures, ensuring consistency and reducing errors in action creation.

Why we use **addCase** in Redux ?

`addCase` in Redux is used to define reducers for specific action types, enabling modular and organized state management.

Why we use payload in Redux ?

Payloads in Redux are used to carry data from the application to the reducers, allowing actions to pass information that will be used to update the state.

What is State in Redux ?

In Redux, the state represents the entire state of the application, stored as a single immutable object, and managed centrally by the Redux store.

What is Action in Redux ?

In Redux, actions are plain JavaScript objects that represent events or intents to change the state of the application. They typically have a `type` field indicating the type of action being performed and an optional `payload` field containing data associated with the action.

What is payload in Redux ?

Payload in Redux is the data attached to an action object, used to carry information from the application to the reducers, influencing how the state is updated.

How to do it in Practical ?

First Step : npm installation

npm install @reduxjs/toolkit

npm install react-redux

**Change Name After Clicking Button (Static)**

In this case I create two Folders Pages and Reducer. In Pages Folder I create a File Profile.jsx which is my actual rendering component. In Reducer folder I create two Files useReducer.jsx and Store.jsx

useReducer.jsx এ আমরা API কল করবো মানে data রাখবো আর Store.jsx এ কল করে globally rendering হবে।

Step1: useReducer.jsx

//It is a function and it access state and action

import {createReducer} from '@reduxjs/toolkit'

const initialState={

    name:"Mousom",

    age:"23",

    status:"Programmer",

    city:"kolkata",

    phone:"9933935760",

    company:"IBM"

  }

// using reducer

// export default(state=initialState,action)=>{

//     if(action.type==='UPDATE\_AGE'){

//         return {

//             ...state,

//             age:action.payload

//         }

//     }

//     if(action.type==='UPDATE\_PHONE'){

//         return {

//             ...state,

//             phone:action.payload

//         }

//     }

//     return state

// }

// use toolkit

export default createReducer(initialState,(builder)=>{

    builder.addCase('UPDATE\_AGE',(state,action)=>{

        state.age=action.payload

    })

    builder.addCase('UPDATE\_NAME',(state,action)=>{

        state.name=action.payload

    })

    builder.addCase('UPDATE\_CITY',(state,action)=>{

        state.city=action.payload

    })

    builder.addCase('UPDATE\_PHONE',(state,action)=>{

        state.phone=action.payload

    })

    builder.addCase('UPDATE\_COMPANY',(state,action)=>{

        state.company=action.payload

    })

})

Step2: Store.jsx

import { configureStore } from '@reduxjs/toolkit';

import useReducer from './useReducer';

export const Store = configureStore({

    reducer: useReducer,

})

Step3: 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 './Reducers/Store';

import { Provider } from 'react-redux';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <Provider store={Store}>

      <App />

    </Provider>

  </React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

Cover <App/> with <Provider>

Step4: Profile.jsx

import React from 'react'

import { useSelector,useDispatch } from 'react-redux'

const Profile = () => {

    const{name,age,city,phone,company}=useSelector((state)=>{

        console.log(state);

        return state

    })

    //initialze the dispatch

    const dispatch=useDispatch()

    //update the age value

    const updateAge=(age)=>{

        //action to dispatch use useDispatch hooks and after gose to the reducer

        dispatch({type:'UPDATE\_AGE',payload:age})

    }

    const updateName=(name)=>{

      dispatch({type:'UPDATE\_NAME',payload:name})

    }

    const updatecity=(city)=>{

      dispatch({type:'UPDATE\_CITY',payload:city})

    }

    const updatephone=(phone)=>{

      dispatch({type:'UPDATE\_PHONE',payload:phone})

    }

    const updatecompany=(company)=>{

      dispatch({type:'UPDATE\_COMPANY',payload:company})

    }

  return (

    <div>

      <h1>My name is {name}</h1>

      <h1>My age is {age}</h1>

      <h1>My city is {city}</h1>

      <h1>My Mobile Number is {phone}</h1>

      <h1>My Company is {company}</h1>

      <button onClick={()=>updateAge(40)}>AgeUpdate</button>

      <button onClick={()=>updateName('Raju')}>NameUpdate</button>

      <button onClick={()=>updatecity('Delhi')}>NameCity</button>

      <button onClick={()=>updatephone('1234567898')}>Mobile Update</button>

      <button onClick={()=>updatecompany('TCS')}>company Update</button>

    </div>

  )

}

export default Profile

In App.js Import only <Profile/>

**Change Name After Clicking Button (Dynamic)**

মানে আমরা যখন নাম বা অন্য কিছু change করবো তখন সেটি চেঞ্জ হয়ে API এর data টা show করবে

**Step1:** Create a File in Reducers folder name will be Action.jsx

**Action.jsx**

import axios from 'axios';

// Fetch Name

export const fetchName = () => {

    return async (dispatch) => {

        try {

            const apiurl = 'https://jsonplaceholder.typicode.com/users'

            const response = await axios.get(apiurl);

            const result = response.data;

            dispatch({ type: "UPDATE\_NAME", payload: result[1].name });

        } catch (error) {

            console.error('Error fetching data:', error);

        }

    }

}

result[1] means Second Array

**Step2:** call fetchName function in your profile.jsx file in updateName function

**//Change Name Dynamic**

  const updateName = async () => {

    dispatch(fetchName()) // Call fetchName() function

  }

**Step3:**

<button onClick={() => updateName()}>NameUpdate</button>

Call updateName() function in button onClick

**51) How to fetch get api in Redux ?**

Step1:

productslice.js: Where I store my api

import { createAsyncThunk, createSlice } from "@reduxjs/toolkit"; // AsyncThunk handle async function in redux , and createSlice slice the state

import axios from "axios";

// Call Api for Show Product

export const showProduct = createAsyncThunk("showProduct", async (\_, { rejectWithValue }) => {

    try {

        const apiurl = "https://fakestoreapi.com/products"

        const response = await axios.get(apiurl);

        console.log("Fetching Product Data", response);

        return response?.data

    } catch (error) {

        console.log("Error Fetching Read user data", error);

        return rejectWithValue(error.response.data);

    }

});

// createSlice area start

const Productslice = createSlice({

    name: "productdetails",

    initialState: {

        products: [],

        loading: false,

        error: null,

    },

    extraReducers: (builder) => {

        builder

            // Show User

            .addCase(showProduct.pending, (state) => {

                state.loading = true; // If promise Pending

            })

            .addCase(showProduct.fulfilled, (state, action) => {

                state.loading = false;

                state.products = action.payload; // If Promise Fullfilled

            })

            .addCase(showProduct.rejected, (state, action) => {

                state.loading = false; // If Prmise Reject

                state.error = action.payload;

            })

    },

});

export default Productslice.reducer;

**Step2:**

**store.js : Where I export my api function**

**import { configureStore } from "@reduxjs/toolkit"; // Import configureStore**

import Productslice from "./Productslice"; // Import Productslice function from productslice.js file

export const store = configureStore({

  reducer: {

    myproduct: Productslice,

  },

});

Step3:Index.js as same as previous process

Ste4:Product.jsx which will be actual rendering in browser

import React, { useEffect } from "react";

import { useDispatch, useSelector } from "react-redux";

import { showProduct } from "../Reducer/Productslice"; // This is the main function which is call your api

const Product = () => {

    const dispatch = useDispatch();

    const { products } = useSelector((state) => state?.myproduct); // myproduct is store reducer name

    useEffect(() => {

        dispatch(showProduct()); // Call showProduct() function

    }, []);

    console.log('jj',products);

    return (

        <>

            <h2>All data</h2>

            <div>

                {products?.map((value) => (

                    <div key={value.id} className="card w-50 mx-auto my-2">

                        <div className="card-body">

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

                        </div>

                    </div>

                ))}

            </div>

        </>

    );

};

export default Product;

App.js will be same as same as previous

**52) How build details page using useParams() in redux ?**

**Step1: prodetails.js**

import { createAsyncThunk, createSlice } from "@reduxjs/toolkit";

import axios from "axios";

// Call Api for Details product

export const detailsproduct = createAsyncThunk("detailsproduct", async (id, { rejectWithValue }) => {

    try {

        const apiurl = `https://dummyjson.com/products/${id}`

        const response = await axios.get(apiurl);

        console.log("Fetching Details data", response);

        return response.data

    } catch (error) {

        console.log("Error Fetching Details data", error);

        return rejectWithValue(error.response.data);

    }

});

// createSlice area start

const prodetails = createSlice({

    name: "productdetails",

    initialState: {

        singleproduct: [],

        loading: false,

        error: null,

    },

    extraReducers: (builder) => {

        builder

            // Details Product

            .addCase(detailsproduct.pending, (state) => {

                state.loading = true;

            })

            .addCase(detailsproduct.fulfilled, (state, action) => {

                state.loading = false;

                state.singleproduct = action.payload;

            })

            .addCase(detailsproduct.rejected, (state, action) => {

                state.loading = false;

                state.error = action.payload;

            })

    },

});

export default prodetails.reducer;

**Step2: store.js** will be as same as previous

**Step3: Details.jsx** which is rendering page

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

import { useDispatch, useSelector } from "react-redux";

import { useParams } from "react-router-dom";

import { detailsproduct } from "../features/prodetails"

import Layout from "../Common/Layout"

const Details = () => {

    const { id } = useParams();

    const dispatch = useDispatch();

    const { singleproduct, loading } = useSelector((state) => state.pdetails);

    useEffect(() => {

        dispatch(detailsproduct(id));

    }, []);

    if (loading) {

        return (

            <div class="spinner-grow text-primary" role="status" style={{ position: 'absolute', top: '50%', left: '50%', transform: 'translate(-50%, -50%)', fontSize:'100000px'}}>

                <span class="sr-only">Loading...</span>

            </div>

        );

    }

    return (

        <>

            <Layout>

                <div style={{ display: 'flex', justifyContent: 'center', alignItems: 'center', height: '100vh' }}>

                    <div style={{ width: '80%', maxWidth: '1200px', display: 'flex', justifyContent: 'center', alignItems: 'center' }}>

                        <div style={{ width: '50%' }}>

                            <div id="productCarousel" className="carousel slide" data-ride="carousel">

                                <ol className="carousel-indicators">

                                    {singleproduct?.images?.map((image, index) => (

                                        <li key={index} data-target="#productCarousel" data-slide-to={index} className={index === 0 ? 'active' : ''}></li>

                                    ))}

                                </ol>

                                <div className="carousel-inner">

                                    {singleproduct?.images?.map((image, index) => (

                                        <div key={index} className={`carousel-item ${index === 0 ? 'active' : ''}`}>

                                            <img src={image} className="d-block w-100" style={{ height: '300px', objectFit: 'cover' }} alt='' />

                                        </div>

                                    ))}

                                </div>

                                <a className="carousel-control-prev" href="#productCarousel" role="button" data-slide="prev">

                                    <span className="carousel-control-prev-icon" aria-hidden="true"></span>

                                    <span className="sr-only">Previous</span>

                                </a>

                                <a className="carousel-control-next" href="#productCarousel" role="button" data-slide="next">

                                    <span className="carousel-control-next-icon" aria-hidden="true"></span>

                                    <span className="sr-only">Next</span>

                                </a>

                            </div>

                        </div>

                        <div style={{ width: '50%', paddingLeft: '20px' }}>

                            <div className="card">

                                <div className="card-body">

                                    <h5 className="card-title" style={{ color: 'blue' }}>Title: {singleproduct.title}</h5>

                                    <h6 className="card-subtitle mb-2 text-muted">Description: {singleproduct.description}</h6>

                                    <p className="card-text">Price: {singleproduct.price}</p>

                                    <p className="card-text">Rating: {singleproduct.rating}</p>

                                    <p className="card-text">Stock: {singleproduct.stock}</p>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

            </Layout>

        </>

    );

};

export default Details;

**All process are as same as previous**

**53) Handle post data in Redux ?**

**Step1:** userdetails.slice page where I call api

import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";

import axios from "axios";

import { toast } from "react-toastify";

// Call Api for Register

export const register = createAsyncThunk("register", async (data, { rejectWithValue }) => {

    try {

        const apiurl = 'https://wtsacademy.dedicateddevelopers.us/api/user/signup'

        const response = await axios.post(apiurl, data);

        if (response && response?.data?.status === 200) {

            console.log("Fetching Register data", response);

            toast.success(response?.data?.message)

            return response.data;

        } else {

            console.log("Error Fetching Register Data", response);

            toast.error(response?.data?.message)

            return response.data;

        }

    } catch (error) {

        console.log("Error Fetching Register data", error);

        toast.error(error?.response?.data?.message)

        return rejectWithValue(error.response.data);

    }

});

// createSlice area start

const userDetailSlice = createSlice({

    name: "userDetail",

    initialState: {

        users: [],

        loading: false,

        error: null,

        getData: [],

    },

extraReducers: (builder) => {

        builder

            // Register

            .addCase(register.pending, (state) => {

                state.loading = true;

            })

            .addCase(register.fulfilled, (state, action) => {

                state.loading = false;

                state.users.push(action.payload);

            })

            .addCase(register.rejected, (state, action) => {

                state.loading = false;

                state.error = action.payload.message;

            })

  },

});

export default userDetailSlice.reducer;

**Step2:** store.js and index.js will be as same as previous

Note: Create different reducer for different API

reducer: {

    app: userDetailSlice, // Put useDetailSlice function in app reducer you put different reducer for different api

  },

**Step3:** Register.jsx actual rendering page

import React, { useState } from 'react'

import Layout from '../Common/Layout'

import { useSelector, useDispatch } from 'react-redux'

import { register } from '../features/userDetailSlice'

import { useNavigate } from 'react-router-dom'

const myobjects = {

    first\_name: '',

    last\_name: '',

    email: '',

    password: '',

    profile\_pic: ''

}

const Register = () => {

    const [reg, setReg] = useState([]);

    const [image, setImage] = useState();

    const { loading } = useSelector((state) => state.app);

    const navigate = useNavigate();

    const dispatch = useDispatch();

    const handleOnChange = (e) => {

        const { name, value } = e.target

        setReg({ ...reg, [name]: value })

    };

    const handleOnSubmit = async (e) => {

        e.preventDefault();

        try {

            // For to Handle Form data

            let formdata = new FormData();

            formdata.append("first\_name", reg.first\_name);

            formdata.append("last\_name", reg.last\_name);

            formdata.append("email", reg.email);

            formdata.append("password", reg.password);

            formdata.append("profile\_pic", image);

            // Because it is a form data so I put formdata in () for raw data you put reg in () which is piller of useState

            const response = await dispatch(register(formdata));

            if (response.payload && response?.payload.status === 200) {

                navigate("/login");

            }

        } catch (error) {

            console.log("Error", error);

        }

    };

    return (

        <>

            <Layout>

                <form method='post' onSubmit={handleOnSubmit} style={{ maxWidth: '400px', margin: 'auto', marginTop: '50px', padding: '20px', border: '1px solid #ccc', borderRadius: '5px', boxShadow: '0 2px 5px rgba(0,0,0,0.1)' }}>

                    <div style={{ marginBottom: '20px' }}>

                        <label htmlFor="exampleInputEmail1" style={{ display: 'block', marginBottom: '5px' }}>First Name</label>

                        <input type="text" className="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" name='first\_name' onChange={handleOnChange} style={{ width: '100%', padding: '10px', borderRadius: '5px', border: '1px solid #ccc' }} />

                    </div>

                    <div style={{ marginBottom: '20px' }}>

                        <label htmlFor="exampleInputEmail1" style={{ display: 'block', marginBottom: '5px' }}>Last Name</label>

                        <input type="text" className="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" name='last\_name' onChange={handleOnChange} style={{ width: '100%', padding: '10px', borderRadius: '5px', border: '1px solid #ccc' }} />

                    </div>

                    <div style={{ marginBottom: '20px' }}>

                        <label htmlFor="exampleInputEmail1" style={{ display: 'block', marginBottom: '5px' }}>Email address</label>

                        <input type="email" className="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" name='email' onChange={handleOnChange} style={{ width: '100%', padding: '10px', borderRadius: '5px', border: '1px solid #ccc' }} />

                    </div>

                    <div style={{ marginBottom: '20px' }}>

                        <label htmlFor="exampleInputPassword1" style={{ display: 'block', marginBottom: '5px' }}>Password</label>

                        <input type="password" className="form-control" id="exampleInputPassword1" name='password' onChange={handleOnChange} style={{ width: '100%', padding: '10px', borderRadius: '5px', border: '1px solid #ccc' }} />

                    </div>

                    {/\*This form section is for the submit image\*/}

                    <div style={{ marginBottom: '20px' }}>

                        <input type="file" onChange={(e) => setImage(e.target.files[0])} name="image" accept="image/\*" className="form-control" />

                        {image !== "" && image !== undefined && image !== null ? (

                            <img style={{ height: "180px" }} src={URL.createObjectURL(image)} alt="" className="upload-img" />

                        ) : (

                            <>{image === "" && <p style={{ color: 'white' }}>Drag or drop content here</p>}</>

                        )}

                    </div>

                    {/\*Image area end\*/}

                    <button type="submit" className="btn btn-primary" style={{ width: '100%', padding: '10px', borderRadius: '5px', border: 'none', background: '#007bff', color: '#fff' }}>

                        {loading ? 'Loading...' : 'Register'}

                    </button>

                </form>

            </Layout>

        </>

    )

}

export default Register

**55) Add to Cart In Redux using Filter**

cartslice.js

import { createSlice } from "@reduxjs/toolkit";

const initialState = [];

const cartSlice = createSlice({

name: 'cart',

initialState,

reducers: {

// For Add to Cart Item

addToCart(state, action) {

const { id, ...item } = action.payload;

const existingItem = state.find(item => item.id === id);

if (existingItem) {

existingItem.quantity++;

} else {

state.push({ id, quantity: 1, ...item });

}

},

// For Remove Cart Item

removeCartItem(state, action) {

return state.filter(item => item.id !== action.payload);

},

// For Increament Quantity

incrementQuantity(state, action) {

const { id } = action.payload;

const item = state.find(item => item.id === id);

if (item) {

item.quantity++;

}

},

// For Decreament Quantity

decrementQuantity(state, action) {

const { id } = action.payload;

const item = state.find(item => item.id === id);

if (item && item.quantity > 1) {

item.quantity--;

}

}

}

});

export const { addToCart, removeCartItem, incrementQuantity, decrementQuantity } = cartSlice.actions;

export default cartSlice.reducer;

Product.jsx

a) import { addToCart } from "../features/cartslice";

b) const handleAddToCart = (product) => {

dispatch(addToCart(product));

};

c) <Button size="small" color="primary" onClick={() => handleAddToCart(value)}>

Add to Cart

</Button>

Cart.jsx: Where product will be store

import React from 'react';

import { useSelector, useDispatch } from 'react-redux';

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

import { removeCartItem, incrementQuantity, decrementQuantity } from '../features/cartslice';

import Layout from '../Common/Layout';

const Cart = () => {

const dispatch = useDispatch();

const cartItems = useSelector((state) => state.cart);

const handleRemoveFromCart = (itemId) => {

dispatch(removeCartItem(itemId));

};

const handleIncrement = (itemId) => {

dispatch(incrementQuantity({ id: itemId }));

};

const handleDecrement = (itemId) => {

dispatch(decrementQuantity({ id: itemId }));

};

const getTotalPrice = () => {

return cartItems.reduce((total, item) => total + item.price \* item.quantity, 0);

};

return (

<Layout>

<div className='container mt-5'>

<h1>Shopping Cart</h1>

{cartItems.length === 0 ? (

<p>Your cart is empty</p>

) : (

<>

<table className="table">

<thead>

<tr>

<th scope="col">Image</th>

<th scope="col">Title</th>

<th scope="col">Description</th>

<th scope="col">Price</th>

<th scope="col">Quantity</th>

<th scope="col">Total</th>

<th scope="col">Actions</th>

</tr>

</thead>

<tbody>

{cartItems.map((item) => (

<tr key={item.id}>

<td><img src={item.thumbnail} alt={item.title} style={{ height: '60px', borderRadius: '5%' }} /></td>

<td>{item.title}</td>

<td>{item.description}</td>

<td>${item.price.toFixed(2)}</td>

<td>{item.quantity}</td>

<td>${(item.price \* item.quantity).toFixed(2)}</td>

<td>

<button className="btn btn-sm btn-primary me-2" onClick={() => handleIncrement(item.id)}>+</button>

<button className="btn btn-sm btn-secondary" onClick={() => handleDecrement(item.id)}>-</button>

<button className="btn btn-sm btn-danger ms-2" onClick={() => handleRemoveFromCart(item.id)}>Remove</button>

</td>

</tr>

))}

</tbody>

</table>

<div className="text-end mb-4">

<strong>Total: ${getTotalPrice().toFixed(2)}</strong>

</div>

<Link to='/'>

<button className="btn btn-primary">Continue Shopping</button>

</Link>

</>

)}

</div>

</Layout>

);

};

export default Cart;

Navbar.jsx

import React from 'react';

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

import { useSelector } from 'react-redux';

const Navbar = () => {

// For to show number of cart

const cartItems = useSelector((state) => state.cart);

const navbarStyle = {

backgroundColor: '#343a40',

padding: '15px 0',

marginBottom: '20px',

};

const linkStyle = {

color: 'white',

textDecoration: 'none',

marginRight: '20px',

};

const cartLinkStyle = {

color: 'white',

textDecoration: 'none',

marginRight: '20px',

position: 'relative',

};

const cartItemCountStyle = {

backgroundColor: 'red',

color: 'white',

borderRadius: '50%',

padding: '5px 10px',

position: 'absolute',

top: '-10px',

right: '-10px',

};

return (

<nav style={navbarStyle}>

<div className="container">

<ul className="list-unstyled d-flex justify-content-between align-items-center mb-0">

<li>

<Link to="/" style={linkStyle}>Home</Link>

</li>

<li>

<Link to="/cart" style={cartLinkStyle}>

Cart ({cartItems.length})

{cartItems.length > 0 && <span style={cartItemCountStyle}>{cartItems.length}</span>}

</Link>

</li>

</ul>

</div>

</nav>

);

};

export default Navbar;

**Note:** In this Navbar there will be show number of cart item in the Cart.jsx file

**56) Authentication Using Redux**

**Step1**

Keep your base url in .env file just like this

# Store Base url here

REACT\_APP\_BASE\_URL = https://restapinodejs.onrender.com/api/

**Step2**

Create axiosInstance in api.js file

// Here I create axiosInstance

import axios from "axios";

let axiosInstance = axios.create({

baseURL: process.env.REACT\_APP\_BASE\_URL,

});

axiosInstance.interceptors.request.use(

async function (config) {

const token =

localStorage.getItem("token") || sessionStorage.getItem("token");

if (token !== null || token !== undefined) {

config.headers["x-access-token"] = token; // Here I pass token

}

return config;

},

function (err) {

return Promise.reject(err);

}

);

export default axiosInstance;

**Step3:** Create Login and Register Normal Form without added functionality

**Step4:** Create a file authslice.js here I call two api’s login and register.

import { createSlice, createAsyncThunk } from '@reduxjs/toolkit'

import axiosInstance from '../api/api'

import { toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css'

const initialState = {

    loading: false,

    user: {}, // for user object

    redirectTo: null, // For Redirect Page

    Logouttoggle: false, // For Logout Button

    userName: false,

    redirectReg: null // For to Redirect Login

}

// API fetch for Register

export const registerUser = createAsyncThunk("/signup", async (user) => {

    try {

        const apiurl = "register"

        const response = await axiosInstance.post(apiurl, user);

        if (response && response?.data?.success === true) {

            toast.success(response?.data?.message)

        } else {

            toast.error(response?.data?.message)

        }

        return response?.data;

    } catch (error) {

        toast.error(error?.response?.data?.msg);

        console.log(error);

    }

});

// API fetch for Login

export const loginRequest = createAsyncThunk("login", async (user) => {

    try {

        const apiurl = "login"

        const response = await axiosInstance.post(apiurl, user);

        if (response && response?.data?.status === 200) {

            toast.success(response?.data?.message)

        } else {

            toast.error(response?.data?.message)

        }

        return response?.data;

    } catch (error) {

        toast.error(error?.response?.data?.message);

        console.log(error);

    }

});

export const AuthSlice = createSlice({

    name: 'user',

    initialState,

    reducers: {

        //check for auth token

        check\_token: (state, { payload }) => {

            let token = localStorage.getItem("token");

            if (token !== null && token !== undefined) {

                state.Logouttoggle = true;

            }

        },

        logout: (state, { payload }) => {

            localStorage.removeItem("token");

            localStorage.removeItem("name");

            toast.success("Logout successfully")

            state.Logouttoggle = false

        },

        // For to go Register page after keeping token in local storage

        RegLog: (state, { payload }) => {

            localStorage.removeItem("name");

            state.Logouttoggle = false

        },

        // Redirect after login

        redirectToo: (state, { payload }) => {

            state.redirectTo = payload

        },

       // Redirect Register page

        redirectTo\_Register: (state, { payload }) => {

            state.redirectReg = payload

        }

    },

    extraReducers: (builder) => {

        // Register Request

        builder

            //For Registration Pending

            .addCase(registerUser.pending, (state) => {

                state.loading = true;

                state.error = null;

            })

            //For Registration Fulfilled

            .addCase(registerUser.fulfilled, (state, { payload }) => {

                state.loading = false;

                if (payload && payload.success === true) { // Check if payload exists

                    localStorage.setItem("name", payload.data.name); // I use .data because data is present in API

                    state.redirectReg = "/login"; // Redirect Login If Get Name in storage

                    // toast.success(`Hi ${payload?.data?.name}, ${payload?.message}`);

                }

            })

            //For Registration Reject

            .addCase(registerUser.rejected, (state, { payload }) => {

                state.loading = false;

                state.error = payload;

            });

        // Login Request

        builder

            .addCase(loginRequest.pending, (state, action) => {

                state.loading = true;

            })

            .addCase(loginRequest.fulfilled, (state, { payload }) => {

                state.loading = false;

                if (payload?.status === 200) {

                    localStorage.setItem("token", payload?.token);

                    localStorage.setItem("name", payload?.user.name); // Here I use .user beacuse in this API user is present not data

                    state.Logouttoggle = true;

                    state.redirectTo = "/blog"; // Redirect Blog After storing data in Local storage

                    // toast.success(`Hi ${payload?.user.name}, ${payload?.message}`);

                }

            })

            .addCase(loginRequest.rejected, (state, action) => {

                state.loading = false;

            });

    }

})

export const {

    check\_token, redirectToo, logout, redirectTo\_Register, RegLog } = AuthSlice.actions

**Step5:** Import authslice function in store like this

import { configureStore } from "@reduxjs/toolkit";

import {AuthSlice} from "../Auth/authslice" // In this case we use { } because we not do export default only do export

export const store = configureStore({

    reducer: {

                Auth: AuthSlice.reducer, // Reducer for Auth

      },

});

After doing this go to index.js and cover <App/> with Provider as same as previous process

**Step6:** Register.jsx

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

import Layout from '../Common/Layout';

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

import { useSelector, useDispatch } from 'react-redux';

import { registerUser } from '../Auth/authslice'; // Import registerUser function

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

import { Typography, Container, Avatar, Grid, CssBaseline, TextField, Button, Box, FormControlLabel, Checkbox, Paper } from '@mui/material';

import AppRegistrationIcon from '@mui/icons-material/AppRegistration';

import Loader from '../Common/Loader'; // Import Loading

const initialValue = {

    name: '',

    email: '',

    mobile: '',

    password: '',

    photo: ''

};

const Register = () => {

    const navigate = useNavigate();

    const dispatch = useDispatch();

    const { redirectReg, loading } = useSelector((state) => state?.Auth); // Get Data from Store

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

    const [error, setError] = useState({});

    const [image, setImage] = useState(null);

    const validation = () => {

        let error = {};

        if (!user.name) {

            error.name = 'Name is Required';

        }

        if (!user.email) {

            error.email = 'Email is Required';

        } else if (!/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(user.email)) {

            error.email = 'Invalid Email';

        }

        if (!user.mobile) {

            error.mobile = 'Mobile is Required';

        }

        if (!user.password) {

            error.password = 'Password is Required';

        }

        return error;

    };

    const postUserData = (e) => {

        const { name, value } = e.target;

        setUser({ ...user, [name]: value });

       // Short Cut Validation process in handle on change

        if (!value) {

            setError({ ...error, [name]: `${name.charAt(0).toUpperCase() + name.slice(1)} is Required` });

        } else {

            setError({ ...error, [name]: '' });

        }

    };

    const SubmitInfo = (e) => {

        e.preventDefault();

        const ErrorList = validation();

        if (Object.keys(ErrorList).length === 0) {

            const formData = new FormData();

            formData.append('name', user.name);

            formData.append('email', user.email);

            formData.append('mobile', user.mobile);

            formData.append('password', user.password);

            formData.append('photo', image);

            dispatch(registerUser(formData));

        }

    };

    // For Redirect which is part of Authentication (Start)

    const redirectUser = () => {

        const name = localStorage.getItem('name');

        const isInLoginPage = window.location.pathname.toLowerCase() === '/register';

        if (name !== null && name !== undefined && name !== '') {

            isInLoginPage && navigate('/login');

        }

    };

    useEffect(() => {

        redirectUser();

    }, [redirectReg]);

    // For Redirect which is part of Authentication (End)

    return (

        <Layout>

            <Container component="main" maxWidth="xs">

                <CssBaseline />

                <Paper elevation={3} sx={{ padding: 4, marginTop: 8 }}>

                    <Avatar sx={{ bgcolor: 'secondary.main', margin: 'auto' }}>

                        <AppRegistrationIcon />

                    </Avatar>

                    <Typography component="h1" variant="h5" align="center" sx={{ mt: 2 }}>

                        Register

                    </Typography>

                    <Box component="form" onSubmit={SubmitInfo} sx={{ mt: 2 }}>

                        <TextField

                            margin="normal"

                            fullWidth

                            id="name"

                            label="Name"

                            name="name"

                            autoFocus

                            value={user.name}

                            onChange={postUserData}

                            error={!!error.name}

                            helperText={error.name}

                        />

                        <TextField

                            margin="normal"

                            fullWidth

                            id="email"

                            label="Email Address"

                            name="email"

                            value={user.email}

                            onChange={postUserData}

                            error={!!error.email}

                            helperText={error.email}

                        />

                        <TextField

                            margin="normal"

                            fullWidth

                            id="mobile"

                            label="Mobile"

                            name="mobile"

                            value={user.mobile}

                            onChange={postUserData}

                            error={!!error.mobile}

                            helperText={error.mobile}

                        />

                        <TextField

                            margin="normal"

                            fullWidth

                            name="password"

                            label="Password"

                            type="password"

                            id="password"

                            value={user.password}

                            onChange={postUserData}

                            error={!!error.password}

                            helperText={error.password}

                        />

                        {/\*This form section is for the submit image\*/}

                        <div style={{ marginBottom: '20px' }}>

                            <input type="file" onChange={(e) => setImage(e.target.files[0])} name="image" accept="image/\*" className="form-control" />

                            {image !== "" && image !== undefined && image !== null ? (

                                <img style={{ height: "180px" }} src={URL.createObjectURL(image)} alt="" className="upload-img" />

                            ) : (

                                <>{image === "" && <p style={{ color: 'white' }}>Drag or drop content here</p>}</>

                            )}

                        </div>

                        {/\*Image area end\*/}

                        <Button type="submit" fullWidth variant="contained" sx={{ mt: 3, mb: 2 }}>

                            {loading? <Loader/> : 'Register'}

                        </Button>

                    </Box>

                    <Grid container justifyContent="flex-end">

                        <Grid item>

                            <Link to="/login" variant="body2">

                                Already have an account? Sign in

                            </Link>

                        </Grid>

                    </Grid>

                </Paper>

            </Container>

        </Layout>

    );

};

export default Register;

**Step7:**Login.jsx

import React from 'react';

import { useState, useEffect } from 'react';

import { useSelector, useDispatch } from 'react-redux';

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

import Layout from '../Common/Layout';

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

import { loginRequest, RegLog } from '../Auth/authslice'; // Import loginRequest function

import Avatar from '@mui/material/Avatar';

import Button from '@mui/material/Button';

import CssBaseline from '@mui/material/CssBaseline';

import TextField from '@mui/material/TextField';

import Grid from '@mui/material/Grid';

import Box from '@mui/material/Box';

import Typography from '@mui/material/Typography';

import Container from '@mui/material/Container';

import LoginIcon from '@mui/icons-material/Login';

import Loader from '../Common/Loader'; // Import Loader

const initialValue = {

    email: '',

    password: ''

}

const Login = () => {

    const navigate = useNavigate();

    const dispatch = useDispatch();

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

    const { redirectTo, loading } = useSelector((state) => state?.Auth);

    const [error, setError] = useState({});

    const handleChange = (e) => {

        const { name, value } = e.target;

        setUser({ ...user, [name]: value });

        setError({ ...error, [name]: '' });

    };

    const handleSubmit = async (e) => {

        e.preventDefault();

        let ErrorList = validation();

        setError(ErrorList);

        if (Object.keys(ErrorList).length === 0) {

            dispatch(loginRequest(user));

        }

    };

    const validation = () => {

        let error = {}

        if (!user.email) {

            error.email = 'Email is required';

        } else if (!/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(user.email)) {

            error.email = 'Invalid email format';

        }

        if (!user.password) {

            error.password = 'Password is required';

        }

        return error;

    };

    //Redirect if get the token or not get the token

    const redirectUser = () => {

        let token = localStorage.getItem("token")

        let isInLoginPage = window.location.pathname.toLowerCase() === "/login";

        if (token !== null && token !== undefined && token !== "") {

            isInLoginPage && navigate("/blog");

        }

    }

    useEffect(() => {

        redirectUser()

    }, [redirectTo])

    // Redirect Area End

    // If I not use this function then I can't go register page when token will be present in local storage

    const log = () => {

        dispatch(RegLog())

    }

    return (

        <Layout>

            <Container component="main" maxWidth="xs" style={{ marginTop: '150px'}}>

                <CssBaseline />

                <Box

                    sx={{

                        marginTop: 8,

                        display: 'flex',

                        flexDirection: 'column',

                        alignItems: 'center',

                    }}

                >

                    <Avatar sx={{ m: 1, bgcolor: 'secondary.main' }}>

                        <LoginIcon />

                    </Avatar>

                    <Typography component="h1" variant="h5">

                        Login

                    </Typography>

                    <Box component="form" onSubmit={handleSubmit} noValidate sx={{ mt: 1 }}>

                        <TextField

                            margin="normal"

                            required

                            fullWidth

                            id="email"

                            label="Email Address"

                            name="email"

                            autoComplete="email"

                            autoFocus

                            value={user.email}

                            onChange={handleChange}

                            error={!!error.email}

                            helperText={error.email}

                        />

                        <TextField

                            margin="normal"

                            required

                            fullWidth

                            name="password"

                            label="Password"

                            type="password"

                            id="password"

                            autoComplete="current-password"

                            value={user.password}

                            onChange={handleChange}

                            error={!!error.password}

                            helperText={error.password}

                        />

                        <Button

                            type="submit"

                            fullWidth

                            variant="contained"

                            sx={{ mt: 3, mb: 2 }}

                        >

                            {loading ? <Loader /> : 'Login'}

                        </Button>

                        <Grid container>

                            <Grid item>

                                <Link to="/register" variant="body2" onClick={log}>

                                    {"Don't have an account? Register Now"}

                                </Link>

                            </Grid>

                        </Grid>

                    </Box>

                </Box>

            </Container>

        </Layout>

    );

};

export default Login;

**Step8:** App.js

import React from 'react'

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

import Home from './Pages/Home'

import Blog from './Pages/Blog'

import Categorydetails from './Pages/Categorydetails'

import Blogdetails from './Pages/Blogdetails'

import Allcourse from './Pages/Allcourse'

import Contact from './Pages/Contact'

import { ToastContainer } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

import { Navigate } from 'react-router-dom'

import Login from './Auth/Login'

import Register from './Auth/Register'

import { check\_token } from './Auth/authslice';

import { useDispatch } from 'react-redux';

import { useEffect } from 'react'

const App = () => {

  const dispatch = useDispatch();

  //check token avable or not

  function PrivateRoute({ children }) {

    const token = localStorage.getItem("token") || sessionStorage.getItem("token");

    return token !== null && token !== undefined ? (

      children

    ) : (

      <Navigate to="/login" />

    );

  }

  const private\_routing = [

    {

      path: '/blog',

      component: <Blog />

    },

    {

      path: '/categorydetails/:id',

      component: <Categorydetails />

    },

    {

      path: '/blogdetails/:id',

      component: <Blogdetails />

    },

    {

      path: '/course',

      component: <Allcourse />

    },

    {

      path: '/contact',

      component: <Contact />

    }

  ]

  const public\_routing = [

    {

      path: '/',

      component: <Home />

    },

    {

      path: '/login',

      component: <Login />

    },

    {

      path: '/register',

      component: <Register />

    }

  ]

  // This step is required for to stop page refreshing problem in logout button

  useEffect(() => {

    dispatch(check\_token())

  }, [])

  return (

    <>

      <ToastContainer />

      <Router>

        <Routes>

          {/\*Area of private routing\*/}

          {public\_routing?.map((routing) => {

            return (

              <>

                <Route path={routing?.path} element={routing?.component} />

              </>

            )

          })}

          {/\*Area of public routing\*/}

          {private\_routing?.map((routing) => {

            return (

              <>

                <Route path={routing?.path} element={<PrivateRoute>{routing?.component}</PrivateRoute>} />

              </>

            )

          })}

        </Routes>

      </Router>

    </>

  )

}

export default App

**Step9:** Nav.jsx for to show Logout Button after Login

import \* as React from 'react';

import { toast } from 'react-toastify';

import AppBar from '@mui/material/AppBar';

import Box from '@mui/material/Box';

import Toolbar from '@mui/material/Toolbar';

import IconButton from '@mui/material/IconButton';

import Typography from '@mui/material/Typography';

import Menu from '@mui/material/Menu';

import MenuIcon from '@mui/icons-material/Menu';

import Container from '@mui/material/Container';

import Avatar from '@mui/material/Avatar';

import Button from '@mui/material/Button';

import Tooltip from '@mui/material/Tooltip';

import MenuItem from '@mui/material/MenuItem';

import MenuBookIcon from '@mui/icons-material/MenuBook';

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

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

import { useSelector, useDispatch } from 'react-redux';

import { logout } from '../Auth/authslice';

const pages = ['Home', 'Blog', 'Course', 'Contact'];

const settings = ['Profile', 'Account', 'Dashboard', 'Logout'];

function ResponsiveAppBar() {

    const dispatch = useDispatch();

    const { Logouttoggle } = useSelector((state) => state?.Auth);

    const name = localStorage.getItem("name");

    const navigate = useNavigate()

    // Handle of Logout

    const handleLogout = () => {

        dispatch(logout())

        navigate("/login")

    }

    const [anchorElNav, setAnchorElNav] = React.useState(null);

    const [anchorElUser, setAnchorElUser] = React.useState(null);

    const handleOpenNavMenu = (event) => {

        setAnchorElNav(event.currentTarget);

    };

    const handleOpenUserMenu = (event) => {

        setAnchorElUser(event.currentTarget);

    };

    const handleCloseNavMenu = () => {

        setAnchorElNav(null);

    };

    const handleCloseUserMenu = () => {

        setAnchorElUser(null);

    };

    return (

        <>

            <AppBar position="fixed">

                <Container maxWidth="xl">

                    <Toolbar disableGutters>

                        <MenuBookIcon sx={{ display: { xs: 'none', md: 'flex' }, mr: 1 }} />

                        <Typography

                            variant="h6"

                            noWrap

                            component={Link}

                            to="/"

                            sx={{

                                mr: 2,

                                display: { xs: 'none', md: 'flex' },

                                fontFamily: 'monospace',

                                fontWeight: 700,

                                letterSpacing: '.3rem',

                                color: 'inherit',

                                textDecoration: 'none',

                            }}

                        >

                            PROJECT

                        </Typography>

                        <Box sx={{ flexGrow: 1, display: { xs: 'flex', md: 'none' } }}>

                            <IconButton

                                size="large"

                                aria-label="account of current user"

                                aria-controls="menu-appbar"

                                aria-haspopup="true"

                                onClick={handleOpenNavMenu}

                                color="inherit"

                            >

                                <MenuIcon />

                            </IconButton>

                            <Menu

                                id="menu-appbar"

                                anchorEl={anchorElNav}

                                anchorOrigin={{

                                    vertical: 'bottom',

                                    horizontal: 'left',

                                }}

                                keepMounted

                                transformOrigin={{

                                    vertical: 'top',

                                    horizontal: 'left',

                                }}

                                open={Boolean(anchorElNav)}

                                onClose={handleCloseNavMenu}

                                sx={{

                                    display: { xs: 'block', md: 'none' },

                                }}

                            >

                                {pages.map((page) => (

                                    <MenuItem key={page} onClick={handleCloseNavMenu}>

                                        <Link to={page === 'Home' ? '/' : `/${page.toLowerCase()}`} style={{ textDecoration: 'none', color: 'inherit' }}>

                                            <Typography textAlign="center">{page}</Typography>

                                        </Link>

                                    </MenuItem>

                                ))}

                            </Menu>

                        </Box>

                        <Typography

                            variant="h5"

                            noWrap

                            component="a"

                            href="#app-bar-with-responsive-menu"

                            sx={{

                                mr: 2,

                                display: { xs: 'flex', md: 'none' },

                                flexGrow: 1,

                                fontFamily: 'monospace',

                                fontWeight: 700,

                                letterSpacing: '.3rem',

                                color: 'inherit',

                                textDecoration: 'none',

                            }}

                        >

                            Project

                        </Typography>

                        <Box sx={{ flexGrow: 1, display: { xs: 'none', md: 'flex' }, justifyContent: 'flex-end' }}>

                            {pages.map((page) => (

                                <Button

                                    key={page}

                                    component={Link}

                                    to={page === 'Home' ? '/' : `/${page.toLowerCase()}`}

                                    sx={{ my: 2, color: 'white', display: 'block' }}

                                >

                                    {page}

                                </Button>

                            ))}

                        </Box>

                        <Box sx={{ flexGrow: 0 }}>

                            {Logouttoggle ? (

                                <>

                                    <Tooltip title="Open settings">

                                        <IconButton onClick={handleOpenUserMenu} sx={{ p: 0 }}>

                                            <Avatar alt="Remy Sharp" src="/static/images/avatar/2.jpg" />

                                        </IconButton>

                                    </Tooltip>

                                    <Menu

                                        sx={{ mt: '45px' }}

                                        id="menu-appbar"

                                        anchorEl={anchorElUser}

                                        anchorOrigin={{

                                            vertical: 'top',

                                            horizontal: 'right',

                                        }}

                                        keepMounted

                                        transformOrigin={{

                                            vertical: 'top',

                                            horizontal: 'right',

                                        }}

                                        open={Boolean(anchorElUser)}

                                        onClose={handleCloseUserMenu}

                                    >

                                        {settings.map((setting) => (

                                            <MenuItem key={setting} onClick={setting === 'Logout' ? handleLogout : handleCloseUserMenu}>

                                                {setting === 'Profile' ? (

                                                    <Typography textAlign="center">Profile: {name}</Typography>

                                                ) : (

                                                    <Typography textAlign="center">{setting}</Typography>

                                                )}

                                            </MenuItem>

                                        ))}

                                    </Menu>

                                </>

                            ) : (

                                <Button component={Link} to="/login" sx={{ color: 'inherit', textDecoration: 'none' }}>

                                    Login

                                </Button>

                            )}

                        </Box>

                    </Toolbar>

                </Container>

            </AppBar>

        </>

    );

}

export default ResponsiveAppBar;

**57) How to Perform Delete in Redux?**

**Step1**

Make Slice for delete

import { createSlice, createAsyncThunk } from "@reduxjs/toolkit"; // createSlice make state slice and createAsyncThunk handle asynconomous function

import axiosInstance from '../api/api'

import { toast } from "react-toastify"; // For Toast Message

// Call Api for Delete User

export const deleteproduct = createAsyncThunk("deleteproduct", async (id, { rejectWithValue }) => {

try {

const apiurl = 'product/remove'

const response = await axiosInstance.post(apiurl, { id });

console.log("Fetching delete data", response);

toast.warning(response?.data?.message)

return response.data

} catch (error) {

console.log("Error Fetching delete data", error);

toast.error(error?.response?.data?.message)

return rejectWithValue(error.response.data);

}

});

Note: In this code I use {id} as a object because in this case delete API is post API. If I will delete API then we put id just like this:`product/remove/${id}`

For Delete extraReducers area is not needed If neeeded then you have to put extraReducers

**Step2**

In showproduct.js file you have to put these codes :

import { deleteproduct } from '../Allreducers/deleteslice' // Import delete function

import Swal from 'sweetalert2'; // Import Sweet Alert

The Instalation of sweetalert2 is npm i react-sweetalert2

// Make Handle For Delete (Start)

const handleDelete = async (id) => {

// For Sweet Alert

const result = await Swal.fire({

title: 'Are you sure?',

text: 'You will not be able to recover this product!',

icon: 'warning',

showCancelButton: true,

confirmButtonColor: '#d33',

cancelButtonColor: '#3085d6',

confirmButtonText: 'Yes, delete it!'

});

if (result.isConfirmed) {

await dispatch(deleteproduct(id));

dispatch(show()); // For useQuery I will do refetch()

// After Deletation Message

Swal.fire(

'Deleted!',

'Your product has been deleted',

'success'

);

}

}

// Make Handle For Delete (End)

**Step3**

In Delete Button Area I will do

<Button onClick={() => handleDelete(value.\_id)}>

Delete

</Button>

**58) How to use API Pagination Using Redux ?** (Api Pagination Means Pagination is set in Api)

**Step1**

totalpage: "", // Make For Pagination (put this in your showproductslice.js file initialstate)

Put this code in your showproduct fullfilled area

.addCase(show.fulfilled, (state, action) => {

state.loading = false;

state.showdata = action.payload.data;

state.totalpage = action.payload.totalPages; // For Pagination

})

Note: totalPages is stay in after first data means response?.data so you can’t put response?.data?.data in the time of api calling .

**Step2**

In Showproduct.jsx file put this code :

import Pagination from '@mui/material/Pagination'; // Import Pagination

const { showdata, totalpage, loading } = useSelector((state) => state.Show);

const [currentPage, setCurrentPage] = useState(1); // State For Pagination

// Handle Make For Pagination

const handleOnChange = (e, pageno) => {

setCurrentPage(pageno);

dispatch(show({ page: pageno, perpage: 10 }));

};

useEffect(() => {

dispatch(show());

}, []);

Step3

Put this code in the last of the main function return statement

<Box display="flex" justifyContent="center" alignItems="center" height="100px">

<Grid justifyContent="center">

{showdata.length !== 0 ? (

<Pagination

count={totalpage}

page={currentPage}

onChange={handleOnChange}

variant="outlined"

shape="rounded"

/>

) : (

<>

</>

)}

</Grid>

</Box>

**Important Note :** Redux এর সময় createAsyncThunk আমরা সব API caller সময়ই করবো কিন্তু createSlice আমরা যেখানে প্রয়োজন সেখানে করবো সব জায়গাতে না করলেও চলবে

Check whole CRUD operation using Redux Toolkit from this link :

<https://github.com/mousomray/crud_auth_usequery_hookform_redux>

Allways do all redux project in this process (Usequery, HookFrom, ReduxToolkit)

**Why we use .env file in React ?**

We use .env files in React to store environment variables like API keys or configuration settings, Base url securely.

**Why we use axiosInstance?**

We use axiosInstance in React for creating a customized instance of Axios with default configurations for making HTTP requests.

mainly আমরা এটা use করবো একজায়গায় base url টা রেখে দিতে যাতে base url পাল্টে গেলে সাব্যয়জায়গায় change না করতে হয় একজায়গায় change করলেই কাজ' হবে

How to do it in practical ?

Step1 : Create a .env file in your project not in src or not in public folder.

# Store product base url here

REACT\_APP\_BASE\_URL = https://dummyjson.com/

Step2 : Create a api.js file in your src

// Here I create axiosInstance

import axios from "axios";

let axiosInstance = axios.create({

  baseURL: process.env.REACT\_APP\_BASE\_URL,

});

axiosInstance.interceptors.request.use(

  async function (config) {

    const token =

      localStorage.getItem("token") || sessionStorage.getItem("token");

    if (token !== null || token !== undefined) {

      config.headers["x-access-token"] = token; // Here I pass token

    }

    return config;

  },

  function (err) {

    return Promise.reject(err);

  }

);

export default axiosInstance;

Step3 : When I fetch Api

import axiosInstance from "../api/api"

// Call Api for Category details

export const cdetails = createAsyncThunk("cdetails", async (category, { rejectWithValue }) => {

    try {

        const apiurl = `products/category/${category}`

        const response = await axiosInstance.get(apiurl);

এই পেজ এ আমরা axios import করবো না তার জায়গায় axiosInstance টা ওই api.js থেকে import করবো আর সেটা দিয়েই কাজ করবো

আর base url বাদে বাকি লেজের portion টা এই পেজের apiurl এ রাখবো

For to show .env file in Github go to gitignore in your project > comment out / delete all .env related thing.

এই axiosInstance এর এই process টা mainly আমরা use করবো redux এর সময়

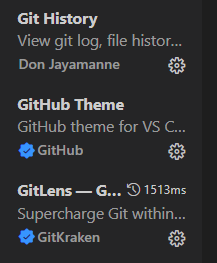
Reactএর সময় আমরা use করবো src={`${process.env.REACT\_APP\_BASE\_URL}blog/image/${item.\_id}`} এই ভাবে

**Last) How to Keep React APP in GitHub ?**

Step1: Download and Install Git bash

<https://git-scm.com/downloads>

Step2: Install these Extensions in your VS code



Last Extention name is GitLens-Git supercharged.

Step3: Create a GitHub repository

Step4 : Right click on your project app folder > click on Open Git Bash here > After open Git Bash put these code

\* git init > Press Enter

\* git add . > Press enter

\* git commit -m “You can put any name” > Press Enter

\* git remote add origin

<https://github.com/mousomray/Klinic_React_Project.git> [Copy this link from your repository and paste it over Git Bash] > Press Enter

\* git push origin master > Press Enter

If you do first time, then one small page will be open > Click on Sign in with your browser > Authorize git-ecosystem .

**How Edit React GitHub repository ?**

Go to your project file > open VS code Terminal > Type these codes

\* git init > Press Enter

\* git add . > Press Enter

\* git commit -m “Create About file in page folder” > Press Enter

\* git push origin master > Press Enter

**How to Collaborate work with another person ?**

Click on settings > Click on collaborators > Add people > Search people > Add people name to this repository > If he change anything I want to see that changes in my VS code go to terminal > git pull origin master > press enter.

How to host react app website in Netlify?

Step1: Create a build file in your react app.

Open your project react app > Open VS code > Go to Terminal > npm run build > Build file will be created.

Step2: Search Netlify in google > Sign-up or login > After login or sign-up go to Sites > browse to upload > Select your project build file which was created earlier > upload.