**Variables and functions**

import { useState } from 'react'

import './App.css'

function App() { //app is a component and component must have return

//creating a state with setter

let [msg,setMsg]= useState('Click me to see something');

//creating a variable

let name="Emma";

let marks=40;

let markForHun = (marks/50)\*100;

//function with parameter

function getMarks(mark){

alert("your mark is "+mark);

console.log("your mark is "+mark); //to get it from console

}

//function without parameter

//getting a variable from the app

//we cannot able to change the value

//first step we have to make it as state and then also have a setter

const clickme = ()=>{

setMsg("You clicked me");

console.log("you enterre1");

}

return(

<div>

<h1>My React App</h1>

<p>Your Name is {name}</p>

<button onClick={()=>{getMarks(markForHun)}}>Get Marks</button>

<hr></hr>

<p>

{

//don't use if else please use ternary operatory

markForHun>80 ? "You are passed with star"

:markForHun>50 ?"You are passed":"you are failed"

}

</p>

<button onClick={()=>{clickme()}}>Click me!!!</button>

<p>{msg}</p>

</div>

)

}

export default App

**UseState Examples**

import { useState } from 'react'

import './App.css'

function App() { //app is a component and component must have return

//creating a state with setter

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

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

const [contact,setContact] = useState(null);

//creating a populate function

const employee=()=>{

setName("John");

setEmail("john@gmail.com");

setContact("9685364589");

}

const manager=()=>{

setName("Albert");

setEmail("albert@gmail.com");

setContact("4569352398");

}

const clear=()=>{

setName("");

setEmail("");

setContact("");

}

return(

<div>

<button onClick={()=>{employee()}}>Employee</button>

<button onClick={()=>{manager()}}>Manager</button>

<p>Name : {name}</p>

<p>Email : {email}</p>

<p>Contact : {contact}</p>

<button onClick={()=>{clear()}}>Clear</button>

</div>

)

}

export default App

**Working with Objects and useEffect()**

import { useEffect, useState } from 'react'

import './App.css'

function App() {   //app is a component and component must have return

  //create a obj for employee and manager

  const [employeeObj,setEmployeeObj] =useState({});

  const [managerObj,setManagerObj] =useState({});

  //create a status

  const [empStatus,setEmpStatus] = useState(false);

  const [managerStatus,setManagerStatus] = useState(false);

  //to assign the value of the object go with useeffect function

  //when you call the use effect function it is going to render the return again

  useEffect(()=>

    {const populate = () => {

      //initialize the employee object

      let empObj={

        "name":"john",

        "email":"emp@gmail.com",

        "contact":"5698743652",

        "projectInfo": "this is the project emp is working on"

      }

      //intialize the manager object

      let manObj={

        "name":"manager",

        "email":"manager@gmail.com",

        "contact":"5698743652",

        "departmentInfo": "IT department",

        "employeeInfo" :"manages 20 employee"

      }

      //assign the value to the objects

      setEmployeeObj(empObj);

      setManagerObj(manObj);

    }

    populate()

  },[]);

const populate=(value)=>{

  switch(value){

    case 'employee':

      setEmpStatus(true);

      setManagerStatus(false);

      break;

    case 'manager':

      setEmpStatus(false);

      setManagerStatus(true);

      break;

    default:

      setEmpStatus(false);

      setManagerStatus(false);

      break;

  }

}

  return(

    <div>

    <button onClick={()=>{populate('employee')}}>Employee</button>

    <button onClick={()=>{populate('manager')}}>Manager</button>

    <div id="employeePopulate">

      {

        empStatus!==true?" ":

        <div>

         <p> "name" : {employeeObj.name}</p>

         <p>  "email" : {employeeObj.email}</p>

         <p> "contact" : {employeeObj.contact}</p>

          <h4>Project Info</h4>

          <span>{employeeObj.projectInfo}</span>

        </div>

      }

    </div>

    <div id="managerPopulate">

      {

        managerStatus !== true?" ":

        <div>

         <p>  "name" : {managerObj.name}</p>

         <p> "email" : {managerObj.email}</p>

         <p> "contact" : {managerObj.contact}</p>

          <h4>Department Info</h4>

          <span>{managerObj.departmentInfo}</span>

          <h4>List of Employee</h4>

          <span>{managerObj.employeeInfo}</span>

        </div>

      }

    </div>

    <hr></hr>

    <br></br>

    <button onClick={()=>{populate('')}}>clear</button>

    </div>

  )

}

export default App

**Using Map to iterate element**

import { useEffect, useState } from "react";

function Post() {

    const [posts, setPosts] = useState([]);

    //hook effect

    useEffect(() => {

        const getPosts = () => {

            let post1 = {

                "userId": 1,

                "id": 1,

                "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",

                "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"

            }

            let post2 = {

                "userId": 1,

                "id": 2,

                "title": "qui est esse",

                "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"

            }

            //save this in the array temp

            let temp = [];

            //push is used to push the object into the temp

            temp.push(post1);

            temp.push(post2);

            //save the temp into the posts

            setPosts(temp);

        }

        //call the getPosts()

        //It is important to call the function useeffect is called itself but

        // function is not.we have to call it

        getPosts();

    }, [])

    return (

        <div>

            <h3>Welcome to Post</h3>

            {/\* to iterate the array we have to use the map and don't use for forEach

            because within the map we can able to write a html but not in forEach

            map syntax is like return syntax \*/}

            {

                posts.map((p, index) => (

                    //it is very importnent to give index or p.id to the div

                    //orelse it will give us error

                    <div key={index}>

                        {p.id} <br />

                        {p.userId} <br />

                        {p.title} <br />

                        {p.body} <br />

                        <hr />

                    </div>

                )

                )

            }

        </div>

    )

}

export default Post;

//jsx = html + css + javascript

/\*\*

 \* DOM: Document Object Model : web browser

 \*

 \* <div>

 \*      <p></p>

 \*      <p></p>

 \* </div>

 \*

 \* body

 \*    div

 \*       p=5655656

 \*       p=565776767

 \*

 \*   React User --> React DOM  --> Browser DOM

 \*  <div>

 \*      <p key={p.id}></p>

 \*      <p key={89898989}></p>

 \* </div>

 \*

 \*/

**Calling the api using fetch**

import { useEffect, useState } from "react";

function Post() {

    const [posts, setPosts] = useState([]);

    //hook effect

    useEffect(() => {

        const getPosts = () => {

            //calling the api

            //fetch the ppost api

            fetch("https://jsonplaceholder.typicode.com/posts")

                //change the response into json

                .then(response => response.json())

                //set the response in the posts

                .then(data => setPosts(data));

        }

        //call the getPosts()

        //It is important to call the function useeffect is called itself but

        // function is not. we have to call it

        getPosts();

    }, [])

    return (

        <div>

            <h3>Welcome to Post</h3>

            {/\* to iterate the array we have to use the map and don't use for forEach

            because within the map we can able to write a html but not in forEach

            map syntax is like return syntax \*/}

            {

                posts.map((p, index) => (

                    //it is very important to give index or p.id to the div

                    //or else it will give us error

                    <div key={index}>

                        {p.id} <br />

                        {p.userId} <br />

                        {p.title} <br />

                        {p.body} <br />

                        <hr />

                    </div>

                )

                )

            }

        </div>

    )

}

export default Post;

**Calling api using axios and delete operation using filter**

* **For using axios we have to install axios**
* **npm i axios or npm install axios**
* **use it in the function**

import axios from "axios";

import { useEffect, useState } from "react";

function Todo() {

    const [todos, setTodos] = useState([]);

    const url = "https://jsonplaceholder.typicode.com/todos";

    useEffect(() => {

        const getTodos = (() => {

            //call the api

            fetch(url)

                .then(response => response.json())

                .then(data => setTodos(data))

        })

        const getTodosWithAxios = async () => {

            try {

                let response = await axios.get(url);

                //console.log(response)

                setTodos(response.data)

            }

            catch (err) {

                console.log(err)

            }

            finally {

            }

        }

        //getTodos();

        getTodosWithAxios()

    }, [])

    const deleteTodo = (todoId) => {

        //console.log(todoId)  -- check and comment

        //make api call to delete -- skip this time

        /\*\*

         \*  delete this todo that the user has clicked from the aaaay

         \* to do this, we must use filter method.

         \*/

        let tempArray = [...todos];

        /\* this step creates a new array tempArry(pos:500X) and takes all elements

        from todos array and saves at new location.

        After this, tempArry will hold all todos.

        Y  is this necessary? todos is a state that we cannot modify. hence we use temp array modify it,

        and using setter give new array to state todo.

        \*/

        tempArray = tempArray.filter(t => t.id !== todoId);

        /\*\*

          \* if this condition 't.id !== todoId' satisfies and gives true then that element stays.

          \* so in short, only element that the user the clicked will leave and others will stay.

          \*/

        setTodos(tempArray);

    }

    return (

        <div>

            <div className="container-fluid">

                <div className="card">

                    <div className="card-header">

                        <h2>Welcome Todo</h2>

                    </div>

                    <div className="card-body">

                        {/\* by using map iterate over the todos \*/}

                        <div className="row ">

                            {

                                todos.map((todo, index) => (

                                    <div className="col-md-4 mb-4" key={index}>

                                        <div className="card">

                                            <div className="card-body">

                                                {todo.id} <br />

                                                {todo.userId} <br />

                                                {todo.title} <br /><br />

                                                <button className="btn btn-danger" onClick={() => { deleteTodo(todo.id) }}>Delete</button>

                                            </div>

                                        </div>

                                    </div>

                                ))

                            }

                        </div>

                    </div>

                </div>

            </div>

        </div>

    )

}

export default Todo;