desctruing

const  data=["uhk","wao","nice"];

         const [a,b,c]=data;

hooks

1)useState

 const [count,setCount]=useState(0);

*return* (

        <div>

         <h1>{a} and {b}  and {c} </h1>

   <h2>{count}</h2>

   <button onClick={()=>setCount(count+1)}> add  num</button>

        </div>

Second example

*import* React,{useState} *from* 'react';

*export* *default* function Statement() {

         const [count,setCount]=useState([{name:"usama",lname:"khan",age:"12"}

      ]);

  const changestate=()=>{

        const users={

              name:"usama hassan khan",

              lname:"khan",

              age:"2o"

        }

        setCount([...count,users]);

  }

*return* (

        <div>{

         count.map((pro)=>{

*return*(

           <div>

           <h1>{pro.name}</h1>

           <h1>{pro.lname}</h1>

           <h1>{pro.age}</h1>

           </div>

              ) ;        })

           }

               <button onClick={changestate}>add</button>

            </div>

        );

}

Use effect

When valle not change it doesnot render

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

const Useeffect = () => {

const [state, setstate] = useState(0);

useEffect(()=>{

console.log("updated ",state)

*return*()=>console.log("i am rerendring");

},[state]

);

return (

<div>

<button onClick={()=>setstate(state+1)}>update</button>

</div>

)

}

export default Useeffect

**here probelms**

here useEffect(()=>{

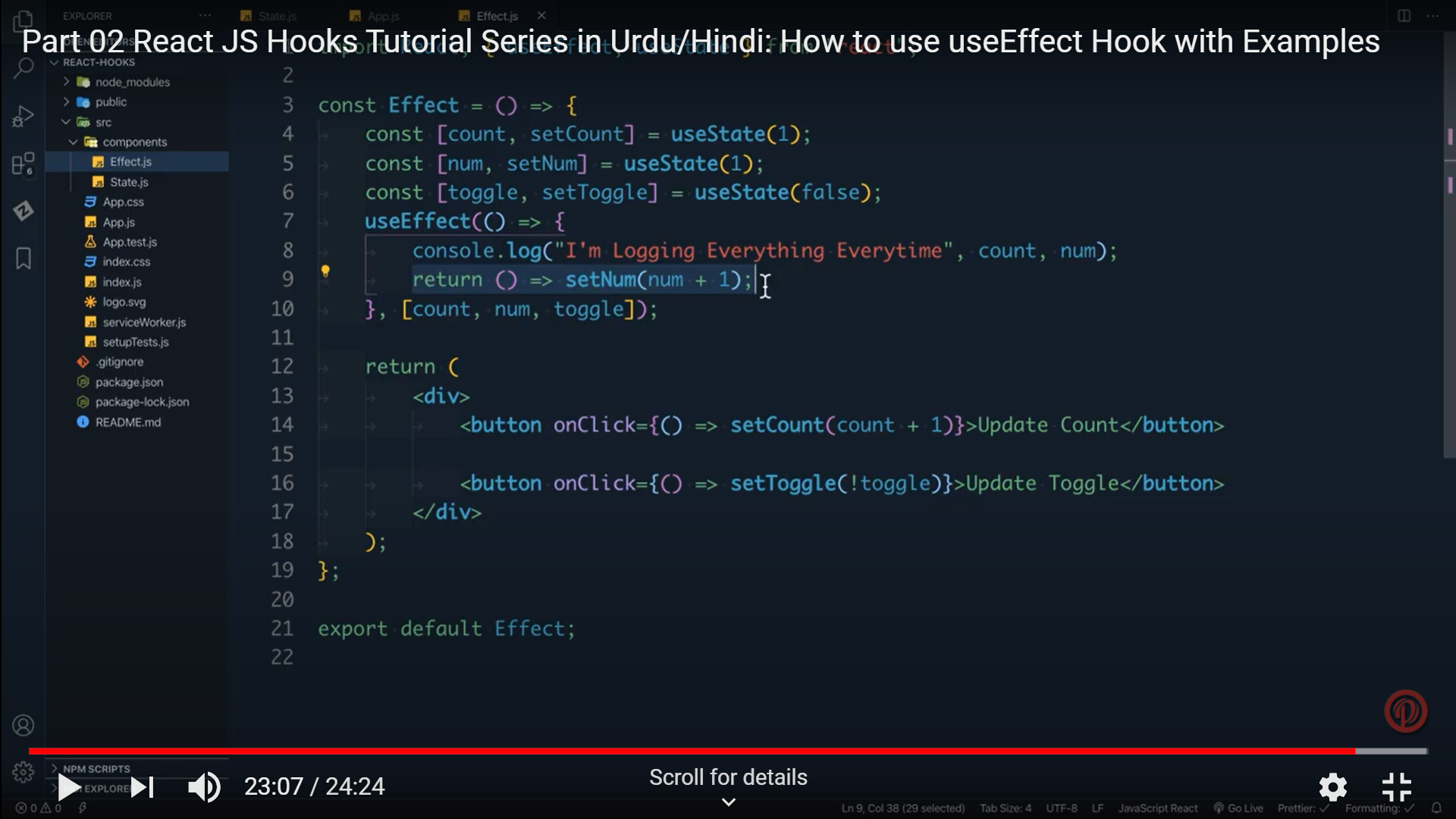
console.log("updated ",state);

*return*()=>console.log("i am rerendring");

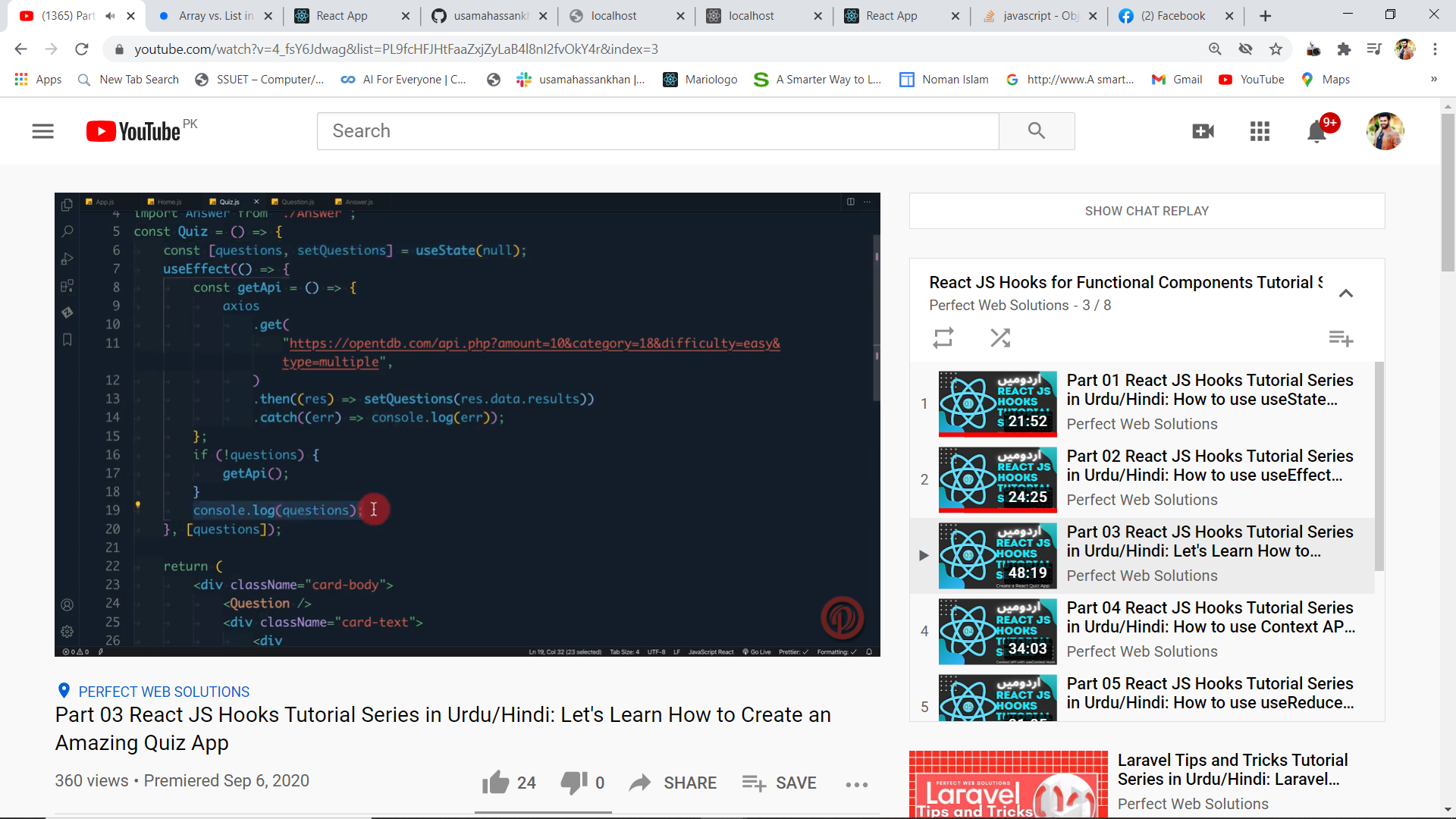
},[state]

);

Return work when cimponed unmount then mount or rerender or remove from vistual dom it happened for micro sec



Quix app



Woking without context appi

**button**

*import* React *from* 'react'

 const Button = ({count,onUpdate}) => {

*return* (

        <div>

           <button onClick={()=>onUpdate(count+1)}>update count</button>

        </div>

    )

}

*export* *default* Button;

message

*import* React *from* 'react'

*export* *default* function message({count}) {

*return* (

  <h2>{count}</h2>

    )

}

App .s

  <Message count={count}/>

     <Button  count={count} onUpdate={ (num)=>setCount(num)}/>

Above example is props now doing context api

Context component

*import* {createContext} *from* 'react';

const Context=createContext({});

*export* *default* Context;

button component

*import* Context *from* './Context';

*import* React ,{useContext}*from* 'react';

const Button = () => {

 const {count,updateState}=useContext(Context);

*return* (

           <button onClick={()=>updateState(count+6)}>update count</button>

    )

}

*export* *default* Button;

message .js

*import* React ,{useContext}*from* 'react';

*import* Button *from* './Button';

*import* Context *from* './Context';

*export* *default* function Message() {

   const {count} = useContext(Context);

*return* (

        <div>

  <h2>{count}</h2>

  <Button/>

  </div>

    );

}

App.js

*import* logo *from* './logo.svg';

*import* './App.css';

*import* React ,{useState} *from* 'react';

*import* State *from* './componenets/Statement'

*import* Useffect *from* './componenets/Useeffect';

*import* Button *from* './componenets/Button';

*import* Message *from* './componenets/Message';

*import* Context *from* './componenets/Context';

function App() {

  const updateState=(num)=>{

    setState(({count,updateState})=>{

*return*{count:num,updateState:updateState};

    });

  };

  const [state,setState]=useState({

count:1,

updateState:updateState,

  });

*return* (

    <div className="App">

     <State/>

     <Useffect/>

    <Context.Provider value={state}>

      <Message/>

    </Context.Provider>

    </div>

  );

}

*export* *default* App;

When u setstate in use it run infine u have to give condition to modify it

Reduce hook

const myReducer =(state,action)=>{

*switch*(action.type){

*case* "check":

*if*((state.count+1)%2==0)*return* {count:state.count+1,oddEven:"Even"};

*else* *return* {count:state.count+1,oddEven:"Odd"};

*default* :

*return* new Error();

    }

};

*export* *default* myReducer;

app.js

*import* './App.css';

*import* React ,{useReducer} *from* 'react';

*import* State *from* './componenets/Statement'

*import* Useffect *from* './componenets/Useeffect';

*import* Button *from* './componenets/Button';

*import* Message *from* './componenets/Message';

*import* Context *from* './componenets/Context';

*import* myReducer *from* './componenets/Reducer';

function App() {

const [state,dispatch]=useReducer(myReducer,{count:1,oddEven:"odd"});

*return*(

  <div>

    <h2>{state.count}</h2>

    <p>{state.oddEven}</p>

    <button onClick={(()=>dispatch({type:"check"}))  }>Check Count</button>

  </div>

);

}

*export* *default* App;

**with extra parameter**

reducer

const myReducer =(state,action)=>{

*switch*(action.type){

*case* "check":

*if*((state.count+1)%2==0)*return* {count:state.count+1,oddEven:"Even"};

*else* *return* {count:state.count+1,oddEven:"Odd"};

*case* "reset":

*return* init();

*default* :

*return* new Error();

    }

};

*export* const init=()=>{

*return* {count:2,oddEven:'Even'};

};

*export* *default* myReducer;

app.js

*import* './App.css';

*import* React ,{useReducer} *from* 'react';

*import* State *from* './componenets/Statement'

*import* Useffect *from* './componenets/Useeffect';

*import* Button *from* './componenets/Button';

*import* Message *from* './componenets/Message';

*import* Context *from* './componenets/Context';

*import*  myReducer,{init} *from* './componenets/Reducer';

function App() {

const [state,dispatch]=useReducer(myReducer,undefined,init);

*return*(

  <div>

    <h2>{state.count}</h2>

    <p>{state.oddEven}</p>

    <button onClick={(()=>dispatch({type:"check"}))  }>Check Count</button>

<button onClick ={()=>dispatch({type:"reset"})}>resets</button>

  </div>

);

}

*export* *default* App;