Task-8

Keeping Components Pure

1. **Convert an impure component that uses Math.random() within the render phase to a pure one.**

**PureComponents.jsx:**

import { useState , useEffect } from "react";

export default function PureComponents(){

    const [randomNumber,setRandomNumber]=useState(0);

    useEffect(()=>{

        setRandomNumber(Math.random())

    },[]);

    return <p>Random Number: {randomNumber}</p>

}

**App.jsx:**

import PureComponents from "./PureComponents.jsx";

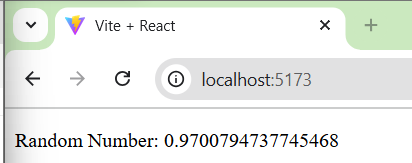
function App() {

  return <PureComponents />;

}

export default App

**Output:**

****

1. **Create a pure component Clock that displays the current time and updates every second without causing side-effects during the render phase.**

**Clock.jsx:**

import { useEffect, useState } from "react";

export default function Clock(){

    const [time,setTime]=useState(new Date());

    useEffect(()=>{

        const timer=setInterval(()=>setTime(new Date()),1000);

        return () => clearInterval(timer);

    },[]);

    return <h3>Time: {time.toLocaleTimeString()}</h3>

}

**App.jsx:**

import Clock from "./Clock.jsx";

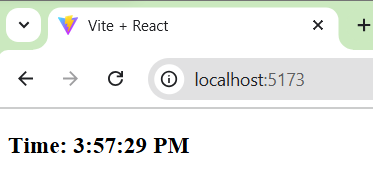
function App() {

  return <Clock />;

}

export default App

**Output:**

****

1. **Use Strict Mode in an existing application and identify any warnings in the console.**

**Clock.jsx:**

import { useEffect, useState } from "react";

export default function Clock(){

    const [time,setTime]=useState(new Date());

    useEffect(()=>{

        const timer=setInterval(()=>setTime(new Date()),1000);

        return () => clearInterval(timer);

    },[]);

    return <h3>Time: {time.toLocaleTimeString()}</h3>

}

**App.jsx:**

import Clock from "./Clock.jsx";

function App() {

  return <Clock />;

}

export default App

**Main.jsx.jsx:**

import { StrictMode } from 'react'

import { createRoot } from 'react-dom/client'

createRoot(document.getElementById('root')).render(

  <StrictMode>

    <App />

  </StrictMode>,

)

1. **Convert a class-based component with side effects in its lifecycle methods to a pure functional component using hooks.**

**Class Components:**

import { Component } from "react";

class Counter extends Component{

    constructor(props){

        super(props);

        this.state={count:0};

    }

    Increament= () => {

        this.setState({count: this.state.count+1})

    }

    render(){

        return(

        <>

          <h2>Count: {this.state.count}</h2>

          <button onClick={this.Increament}>increase</button>

        </>

    )}

}

export default Counter;

**App.jsx:**

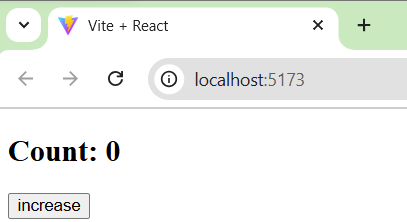
import Counter from "./Count.jsx";

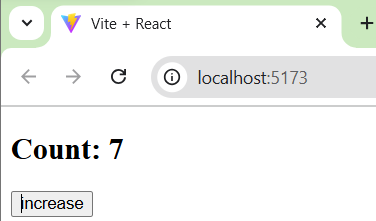
function App() {

  return <Counter />

}

export default App

**Output:**

****

**Functional Components:**

**Counter.jsx:**

import { useState } from "react";

function Counter(){

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

    return(

        <>

        <h2>Count: {count}</h2>

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

        </>

    )

}

export default Counter;

**App.jsx:**

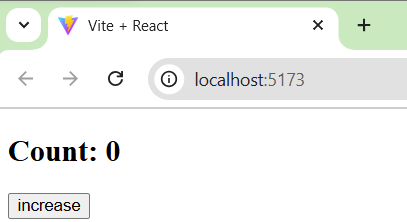
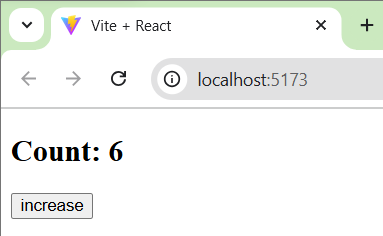
import Counter from "./Count.jsx";

function App() {

  return <Counter />

}

export default App

**Output:**

1. **Make a pure ProfilePic component that takes a user ID as a prop and fetches the user’s profile picture URL from an array without side-effects during rendering.**

**ProfilePic.jsx:**

export default function ProfilePic(props){

    const users=[

        {id: 1, pic:"https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ90Rdf6mqhLFYOYZ8rQ3cvyKuy4ivyndKLgfzpj1AYfVCeyHQ7gRrjhx7jXG\_E37BtUQE&usqp=CAU"},

        {id:2 , pic:"https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSjv-LlSn3OR47vA5HF\_uL2jN2ha-9ZymPMzA&s"}

    ]

    const user = users.find(u => u.id === props.userid )

    return (

        <>

        {user ?(

        <div>

        <h1>Userid:{props.userid}</h1>

        <img style={{display:'block'}} src={user.pic} alt="Profile Picture" />

        </div>

        ):  ( <p>UserId Not Found</p>)}

        </>

    )

}

**App.jsx:**

import ProfilePic from "./ProfilePic.jsx";

function App() {

  return (

    <>

    <ProfilePic userid={1} />

    <ProfilePic userid={2} />

    <ProfilePic userid={3} />

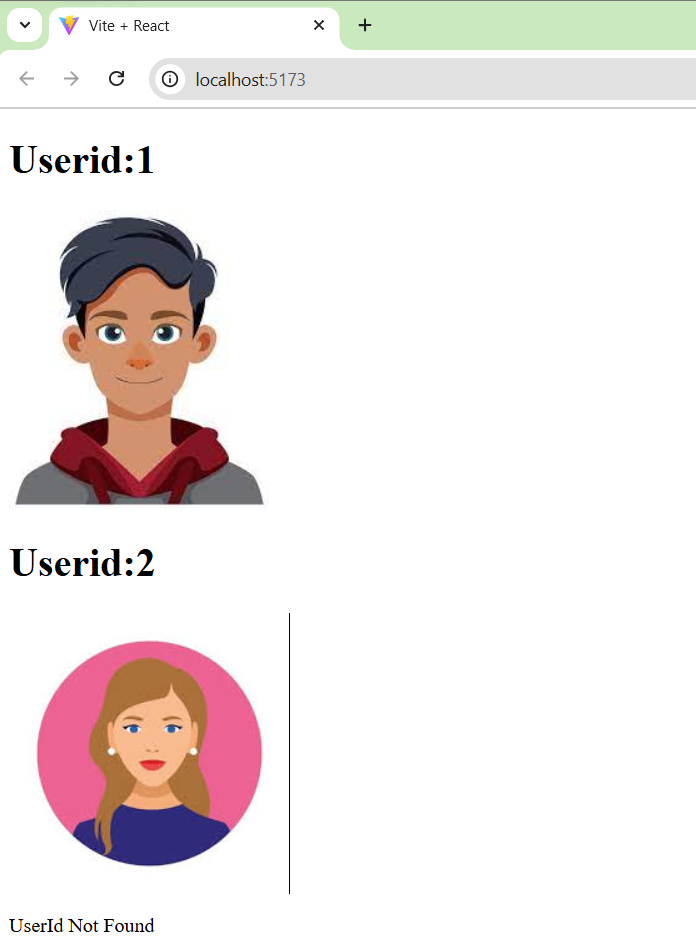
    </>

  );

}

export default App

**Output:**

****