**Practical No.: 06**

**Aim: Using useEffect & useRef Hook**

1. Countdown timer with useEffect

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

function P6i() {

const [timeRemaining, setTimeRemaining] = useState('10');

useEffect(() => {

if (timeRemaining <= 0) {

return;

}

const timerId = setInterval(() => {

setTimeRemaining((prevTime) => prevTime - 1);

console.log('remaining time',timeRemaining )

}, 1000);

return () => {

console.log('Clearing up the time', timeRemaining)

clearInterval(timerId);

};

}, [timeRemaining]);

return (

<div>

<p>Time Remaining: {timeRemaining}</p>

</div>

)

}

export default P6i

**Output:**

**A screen shot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**ii. Window Resize Listener using useEffect**

**Create a WindowSize component that listens to the window's resize event and displays the current window size. Use the useEffect hook to add and remove the event listener on mount and unmount, respectively.**

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

function P6ii() {

const [windowSize, setWindowSize] = useState({ width: window.innerWidth, height: window.innerHeight });

useEffect(() => {

const handleResize = () => {

setWindowSize({ width: window.innerWidth, height: window.innerHeight });

};

window.addEventListener('resize', handleResize);

return () => {

window.removeEventListener('resize', handleResize);

};

}, []);

return (

<div>

<p>Window size: {windowSize.width} x {windowSize.height}</p>

</div>

)

}

export default P6ii

**Output:**

A screenshot of a computer

Description automatically generated

A black and white text

Description automatically generated with medium confidence

iii. Form Input Validation with useEffect

Create a ValidatedInput component that validates user input and shows an error message if the input is invalid. Use the useEffect hook to perform validation whenever the input value changes, simulating componentDidUpdate behavior

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

const FormValidationComponent = () => {

const [inputValue, setInputValue] = useState('');

const [isValid, setIsValid] = useState(true);

useEffect(() => {

console.log('Input value changed:', inputValue);

const isValidInput = inputValue.length > 5;

setIsValid(isValidInput);

}, [inputValue]);

const handleInputChange = (event) => {

setInputValue(event.target.value);

};

const handleSubmit = (event) => {

event.preventDefault();

if (isValid) {

console.log('Form submitted with valid input:', inputValue);

} else {

console.log('Form submission blocked due to invalid input:', inputValue);

}

};

return (

<div>

<form onSubmit={handleSubmit}>

<label>Enter Value:</label>

<input

type="text"

value={inputValue}

onChange={handleInputChange}

style={{ borderColor: isValid ? 'green' : 'red' }}

/>

<button type="submit">Submit</button>

</form>

{!isValid && <p style={{ color: 'red' }}>Input must have more than 5 characters.</p>}

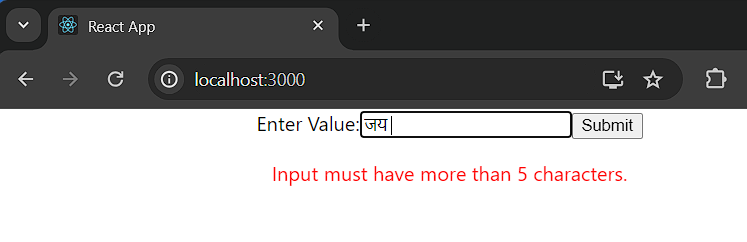
</div>

);

};

export default FormValidationComponent;

**Output:**

****

**A screenshot of a computer

Description automatically generated**

iv. Data Fetching using useEffect

Create a UserPosts component that fetches an jid displays a list of posts using the JSONPlaceholder API (<https://jsonplaceholder.typicode.com/>).

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

function P6iv() {

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

useEffect(() => {

const fetchData = async () => {

const response = await fetch(`https://jsonplaceholder.typicode.com/posts`);

const data = await response.json();

setPosts(data);

};

fetchData();

}, [ ]);

return (

<div>

{posts.map((post) => (

<div key={post.id}>

{post.id}<br></br>

Title: <h3>{post.title}</h3>

Body: <p>{post.body}</p>

</div>

))}

</div>

)

}

export default P6iv

**Output:**

**A screenshot of a computer

Description automatically generated**