**Batch: A2 Roll No.: 16010121045**

**Experiment No. 3**

|  |
| --- |
| **Title:**  React Hooks |

Implementation of React Hooks with the following points

useState

useEffect

useContext

useRef

useCallback

useReducer

useMemo

Code:

App.jsx

*import* React, {

useState,

useEffect,

useRef,

useCallback,

useReducer,

useMemo,

} *from* 'react';

*import* UserInfo *from* './UserInfo';

*import* UserContext *from* './UserContext';

*// Reducer function*

const counterReducer = (state, action) => {

*switch* (action.type) {

*case* 'INCREMENT':

*return* { count: state.count + 1 };

*default*:

*return* state;

}

};

function Counter({ onIncrement }) {

const [state, dispatch] = useReducer(counterReducer, { count: 0 });

const inputRef = useRef(null);

useEffect(() => {

document.title = `Count: ${state.count}`;

inputRef.current.focus();

}, [state.count]);

const handleIncrement = useCallback(() => {

dispatch({ type: 'INCREMENT' });

onIncrement();

}, [onIncrement]);

const squaredCount = useMemo(() => {

*return* state.count \*\* 2;

}, [state.count]);

const [name, setName] = useState('Pargat');

*return* (

<div>

<UserContext.Provider *value*={{ name, setName }}>

<UserInfo />

</UserContext.Provider>

<p>Count: {state.count}</p>

<p>Squared Count: {squaredCount}</p>

<button *onClick*={handleIncrement} *ref*={inputRef}>

Increment

</button>

</div>

);

}

*export* *default* Counter;

UserContext.jsx

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

const UserContext = createContext({

name: null,

setName: () => {},

});

*export* *default* UserContext;

UserInfo.jsx

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

*import* UserContext *from* './UserContext';

function UserInfo() {

const { name } = useContext(UserContext);

*return* <p>User: {name}</p>;

}

*export* *default* UserInfo;

Output:

A screenshot of a computer

Description automatically generated

