**CONTEXT API**

**STEP 1 :- [ create a context object ]**

import React, {createContext,useState,useReducer } from "react";

export const CounterContext = createContext();

**STEP 2 :- [ create a context provider ]**

export const CounterContextProvider = props => {

// initil state

initialstate = {

count:0

}

const [state,dispatch] = useReducer(reducername,initialstate)

return (

<CounterContext.Provider value={{ count,pass all actions }}>

{props.children}

</CounterContext.Provider>

);

};

Export default CounterContextProvider;

**STEP 3 :- [ use context provider ]**

import { CounterContext } from "../context/counter-context";

export default function CounterDisplay() {

const [count] = useContext(CounterContext);

return (

<Statistic>

<Statistic.Value>{count}</Statistic.Value>

<Statistic.Label>Counter</Statistic.Label>

</Statistic>

);

}

**Reducer & Action :-**

const reducer = (state, action) => {

switch (action.type) {

case "ADD\_CONTACT":

return {

contacts: [...state.contacts, action.payload]

};

case "DEL\_CONTACT":

return {

contacts: state.contacts.filter(

contact => contact.id !== action.payload

)

};

case "START":

return {

loading: true

};

case "COMPLETE":

return {

loading: false

};

default:

throw new Error();

}

};