

React Interview Programs

1. useState Increment and Decrement with validation

```
import { useState } from "react";
import "./style.css";
const App = () => {
  const initialvalueofcount = 0;
  const [count, setCount] = useState(0);
  const [formValues, setFormValues] = useState(initialvalueofcount);
  function Increment() {
    if (count <= 10) setCount(count + 1);
  }
  function Decrement() {
    if (count > 0) setCount(count - 1);
  }
  function Clear() {
    setCount(0);
  }
  return (
    <div className="App">
      <button onClick={() => { setCount(count + 1); }}> Increment</button>
      {count}
      <button onClick={Decrement}>Decrement</button>
      <button onClick={Clear}>Clear</button>
    </div>
  );
}; export default App;
```

2. Using useEffect API Data Fetch

```
import React, { useState, useEffect } from "react";
import axios from "axios";
function App() {
  const [data, setData] = useState([]);
  const getData = async () => {
    let response = await axios.get("https://randomuser.me/api");
    console.log("result", response.data);
    setData(response.data);
  };
  useEffect(() => {
    getData();
  }, [data]);
  return (
    <div>
      {
        data &&
```

```

        data.length > 0 && data.map((item) =>{
            return <h4>{item.email}</h4>
        })
    }
    </div>
);
}export default App;

```

3. useContext Program example:

```

import React, { createContext } from "react";
const UserContext = createContext();
const App = () => {
    return (
        <div>
            <center>
                <UserContext.Provider value={"Telugu Skills"}>
                    <ComponentC />
                </UserContext.Provider>
            </center>
        </div>
    );
};
export default App;

```

```

const ComponentC = () => {
    return (
        <div>
            <UserContext.Consumer>
                {(value) => <div>{value}</div>}
            </UserContext.Consumer>
        </div>
    );
};

```

4. ContextAPI Program Example:

5. how i pass data two components and update (increment)the data using contextApi

```

import React, { createContext, useContext, useState } from "react";
export const store = createContext();
const App = () => {
    const [data, setData] = useState(0);
    return (
        <div>
            <center>
                <store.Provider value={[data, setData]}>
                    <ComponentB />
                    <ComponentA />
                    <button onClick={() => setData(data + 1)}>Increment</button>
                </store.Provider>
            </center>
        </div>
    );
};

```

```

export default App;

const ComponentA = () => {
  const [data, setData] = useContext(store);
  return <div>ComponentA{data}</div>;
};
const ComponentB = () => {
  const [data, setData] = useContext(store);
  return <div>ComponentB{data}</div>;
};

```

6. ContextAPI Program Example: how i pass data two components and update the data using contextApi

```

import React, { createContext, useState } from "react";
const UserContext = createContext();
const App = () => {
  const [data, setData] = useState({
    name: "Telugu Skills"
  });
  const handleChange = () => {
    setData({ name: "guna" });
  };
  return (
    <div>
      <center>
        <UserContext.Provider value={data.name}>
          <ComponentC />
          <ComponentB />
          <button onClick={handleChange}>Click</button>
        </UserContext.Provider>
      </center>
    </div>
  );
};
export default App;
const ComponentC = () => {
  return (
    <div>
      <UserContext.Consumer>
        {(value) => <div>{value}</div>}
      </UserContext.Consumer>
    </div>
  );
};
const ComponentB = () => {
  return (
    <div>
      <UserContext.Consumer>
        {(value) => <div>{value}</div>}
      </UserContext.Consumer>
    </div>
  );
};

```

```
);
};
```

7. useRef() Program Example:

```
import { useRef } from "react";
export default function App() {
  const data = useRef(null);
  const submitHandler = (e) => {
    e.preventDefault();
    document.write(data.current.value);
  };
  useEffect(() => {
    data.current.focus();
  }, []);
  return (
    <div>
      <center>
        <form onSubmit={submitHandler}>
          <input ref={data} name="fname" type="text" placeholder="Enter you name" /> <br />
          <button>Submit</button>
        </form>
      </center>
    </div>
  );
}
```

8. UseReducer() Program Example:

```
import React, { useReducer } from "react";
const initialState = { count: 0 };
function reducer(state, action) {
  switch (action.type) {
    case `increment`:
      return { count: state.count + 1 };
    case `decrement`:
      return { count: state.count - 1 };
    default:
      throw new Error();
  }
}
const App = () => {
  const [state, dispatch] = useReducer(reducer, initialState);
  return (
    <div>
      <center>
        <button onClick={() => dispatch({ type: "decrement" })}>
          decrement
        </button>
      </center>
    </div>
  );
}
```

```

    </button>
    Count:{state.count}
    <button onClick={() => dispatch({ type: "increment" })}>
      Increment
    </button>
  </center>
</div>
);
};
export default App;

```

9. UseMemo() Program Example:

```

import React, { useMemo, useState } from "react";

const App = () => {
  const [count, setCount] = useState(0);
  const [number, setNumber] = useState(5);
  const Factorial = useMemo(() => {
    fact(number);
  }, [number]);
  return (
    <>
      Factorial:{Factorial}
      <br />
      <button onClick={() => setCount(count + 1)}>Increment</button>
      count:{count}
    </>
  );
};

const fact = (n) => {
  let answer = 1;
  for (var i = n; i > 1; i--) {
    answer = answer * i;
  }
  console.log("factorial function calling");
  return answer;
};
export default App;

```

10. Redux:

```

import React from 'react';
import { connect } from 'react-redux';
import { IncAction } from './Actions';
import { DecAction } from './Actions';
const App = ({local-variable, IncAction, DecAction})=>{
  return(
    <div>
      <center>
        <button onClick={()=>IncAction(1)}>Increment</button>
        <h1>{local-variable}</h1><br/>
        <button onClick={DecAction}>Decrement</button>

```

```

    </center>
  </div>

)
}
export default App;
const mapStateToProps=state=>({
  local-variable:state
})
Action.js
export const IncAction = (value) => async (dispatch) => {
  dispatch({
    type: "Increment",
    payload: value
  });
};
export const DecAction = () => async (dispatch) => {
  dispatch({
    type: "Decrement"
  });
};
Reducer.js
const count = 0;
export default function reducer(state = count, action) {
  const { type, payload } = action;
  switch (type) {
    case "Increment":
      return state + 1;
    case "Decrement":
      return state - 1;
    default:
      return state;
  }
}
Store.js
import { CreateStore, applyMiddleware } from "redux";
import { componentWithDevTools } from "redux-devtools-extension";
import thunk from "redux-thunk";
import reducer from "./reducer";
const middleware = [thunk];
const store = CreateStore(
  reducer,
  componentWithDevTools(applyMiddleware(...middleware))
);

export default store;

```

11. 1.API Data Fetch

```

import React, { useState, useEffect } from "react";
import axios from "axios";

```

```

function App() {
  const [data, setData] = useState([]);
  const getData = async () => {
    let response = await axios.get("https://randomuser.me/api");
    console.log("result", response.data);
    setData(response.data);
  };
  useEffect(() => {
    getData();
  }, [data]);
  return (
    <div>
      {
        data &&
        data.length > 0 && data.map((item) =>{
          return <h4>{item.email}</h4>
        })
      }
    </div>
  );
}export default App;

```

12. How to update a component every second?

```

import { useState } from "react";
export default function App() {
  const [counter, setCounter] = useState(0);
  const incrementCounter = () => setCounter(counter + 1);
  let decrementCounter = () => setCounter(counter - 1);

  if (counter <= 0) {
    decrementCounter = () => setCounter(0);
  }

  return (
    <div>
      <button onClick={incrementCounter}>inc</button>
      {counter}
      <button onClick={decrementCounter}>dec</button>
    </div>
  );
}

```

13. setTime interval with increment and decrement

```

(function() {
  var update = document.getElementById("liveUpdate");
  var count = {
    digit: 0,
    increment: function() {
      var interval = setInterval(function() {
        if (++count.digit == 10) {
          clearInterval(interval);

```

```

        count.decrement();
      }
      update.innerHTML = count.digit;
    }, 500);
  },
  decrement: function() {
    var interval = setInterval(function() {
      if (--count.digit == -1) {
        clearInterval(interval);
      }
      update.innerHTML = count.digit;
    }, 500);
  }
};
count.increment();
})();

```

14. Increment and Decrement with validation program in react js

```

import { useState } from "react";
import "./style.css";

const App = () => {
  const initialvalueofcount = 0;
  const [count, setCount] = useState(0);
  const [formValues, setFormValues] = useState(initialvalueofcount);

  function Increment() {
    if (count <= 10) setCount(count + 1);
  }
  function Decrement() {
    if (count > 0) setCount(count - 1);
  }
  function Clear() {
    setCount(0);
  }

  return (
    <div className="App">
      <button
        onClick={() => {
          setCount(count + 1);
        }}
      >
        Increment
      </button>
      {count}
      <button onClick={Decrement}>Decrement</button>
      <button onClick={Clear}>Clear</button>
    </div>
  );
};

```



```
export default App;
```

15. #1. Using Flexbox

You can center a div horizontally and vertically with Flex box, with just 4 lines of code.

```
div { display: flex; justify-content: center; align-items: center; height: 100vh; }
```

16. #2. Using Grid

Centering a div with a grid is much easier than with flexbox.

```
div { display: grid; place-items: center; height: 100vh; }
```

17. #3. Using Position Absolute:

You can center a div with CSS positioning too.

```
div { position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); }
```

18. [React Hook : Send data from child to parent component](#)

```
const { useState } = require("react");

export default function PageComponent() {
  const [count, setCount] = useState(0);
  const increment = () => {
    setCount(count + 1);
  };

  return (
    <div className="App">
      <ChildComponent onClick={increment} count={count} />
      <h2>count {count}</h2>
    </div>
  );
}

const ChildComponent = ({ onClick, count }) => {
  return <button onClick={onClick}>Click me {count}</button>;
};
```

19. Props with parent to child

```
export default function App() {
  const arr = [0, 3, 4, 5];
  function add(a, n) {
    return a + n;
  }
}
```

```

const sum = arr.reduce(add, 0);
console.log(sum);
return (
  <div>
    <Welcome name="Sara" />
    <Welcome name="Cahal" />
    <Welcome name="Edite" />
  </div>
);
}
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}

```

```

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

```

```

const formReducer = (state, event) => {
  return {
    ...state,
    [event.name]: event.value
  };
};

```

```

function App() {
  const [formData, setFormData] = useReducer(formReducer, {});
  const [submitting, setSubmitting] = useState(false);

```

```

  const handleSubmit = (event) => {
    event.preventDefault();
    setSubmitting(true);

```

```

    // setTimeout(() => {
    //   setSubmitting(false);
    // }, 3000);
  };

```

```

  const handleChange = (event) => {
    setFormData({
      name: event.target.name,
      value: event.target.value
    });
  };

```

```

  return (
    <div className="wrapper">
      {submitting && (
        <div>
          <ul>
            {Object.entries(formData).map(([name, value]) => (
              <li key={name}>
                <strong>{name}</strong>:{value.toString()}

```

```

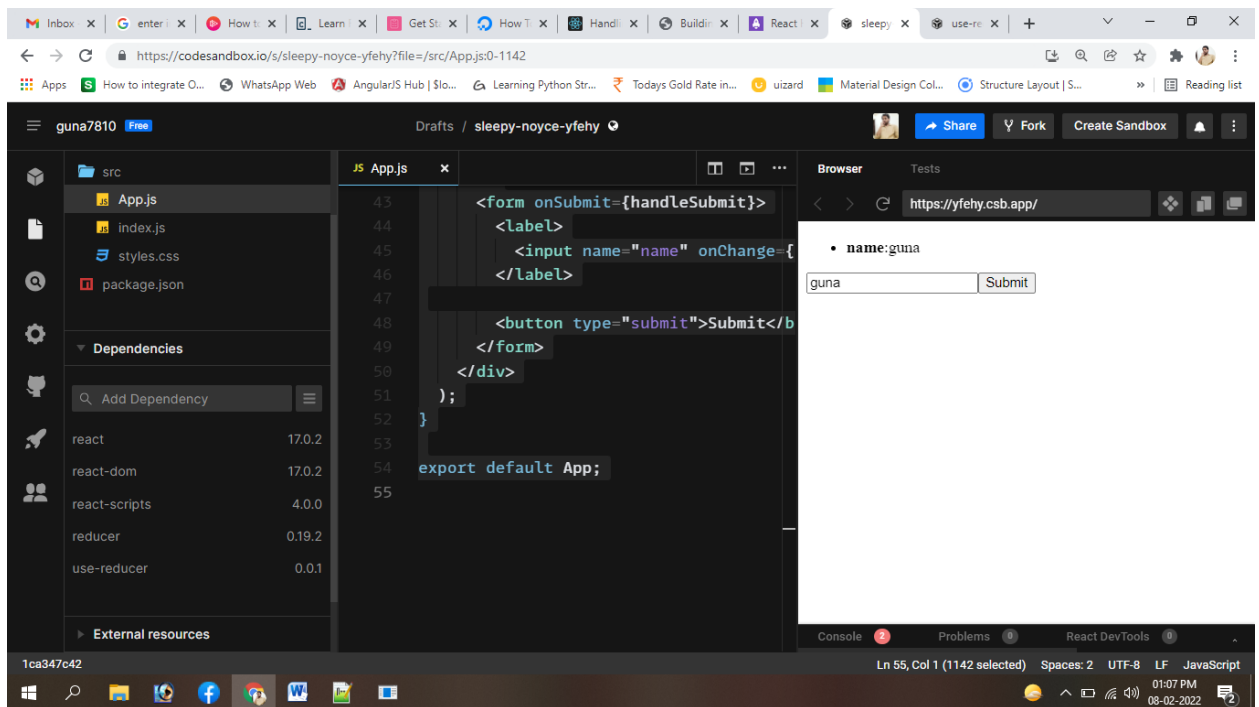
    </li>
  )}
</ul>
</div>
)}
<form onSubmit={handleSubmit}>
  <label>
    <input name="name" onChange={handleChange} />
  </label>

  <button type="submit">Submit</button>
</form>
</div>
);
}

export default App;

```

output:



```

31.<!DOCTYPE html>
<html>
<head>
<style>
div:nth-child(3)>p:nth-child(2){

```

```

    background-color:red;
}
</style>
</head>
<body>
<div>
    <p>hi</p>
    <p>hi</p>
    <p>hi</p>
    <p>hi</p>
</div>
<div>
    <p>hi</p>
    <p>hi</p>
    <p>hi</p>
    <p>hi</p>
</div>
<div>
    <p>hi</p>
    <p>hi</p>
    <p>hi</p>
    <p>hi</p>
</div>
</body>
</html>

```

20. Reactanglebox set all sides with browser

```

<div id="parent">
    <p>First</p>
    <p>Second</p>
    <p>Third</p>
    <p>Fourth</p>
    <p>Five</p>
</div>
#parent {
    position:relative;
    border:3px solid blue;
    height:300px;
    width:500px;
    padding:0;
}
p {
    position:absolute;
    border:2px solid;
    margin:0;
    padding:5px;
    height:10%;
    width:10%;
}
p:nth-child(1) {

```

```

border-color:green;
top:0;
left:0;
}
p:nth-child(2) {
border-color:red;
top:0;
right:0;
}
p:nth-child(3) {
border-color:yellow;
bottom:0;
left:0;
}
p:nth-child(4) {
border-color:pink;
bottom:0;
right:0;
}
p:nth-child(5) {
border-color:black;
top:40%;
right:40%;
transform:translate(-20%);
}

```

21. 1.select box function with parent to child using props?

App.js

```

import React from "react";
import Child from "./Child";

```

```

const App = () => {
  const Cars = [
    {
      name: "BMW",
      value: "110"
    },
    {
      name: "SWIFT",
      value: "120A"
    },
    {
      name: "TOYOTA",
      value: "AP340"
    },
    {
      name: "VOLVO",
      value: "QWE234"
    },
    {

```

```

      name: "RENTAX",
      value: "QW234"
    }
  ];
  return (
    <div className="App">
      <Child data={Cars} />
    </div>
  );
};
export default App;

```

Child.js

```

import React, { useState } from "react";

const Child = ({ data }) => {
  const [value, setValue] = useState("");
  const changeHandle = (e) => {
    setValue(e.target.value);
    console.log(e.target.value);
  };
  return (
    <>
      <select onChange={changeHandle}>
        {data.map((item, index) => {
          return <option value={item.name}>{item.name}</option>;
        })}
      </select>
      <p>Seleted Cars:{value}</p>
    </>
  );
};
export default Child;

```

OUTPUT:

SWIFT ▼

Seleted Cars:SWIFT

22. 2. Only last Name Updated,Don't update First Name ?

```

import React, { useState } from "react";
function App() {
  const [data, setData] = useState({
    FirstName: "",
    LastName: ""
  });
  const changeHandler = (e) => {
    if ("LastName" === e.target.name) {
      setData({ LastName: e.target.value });
    }
  };

```

```

    } else if ("FirstName" === e.target.name) {
      setData({ FirstName: e.target.value });
    }
    console.log("ghgh", data);
  };
  return (
    <div className="App">
      <form>
        <label>FirstName:</label>
        <input type="text" name="FirstName" onChange={changeHandler} />
        <label>LastName:</label>
        <input type="text" name="LastName" onChange={changeHandler} />
      </form>
    </div>
  );
}
export default App;

```

OUTPUT:

FirstName: LastName:

Write a program in React Two Input Field show the data in table form

```

import React, { useState, useEffect } from "react";
// import "./styles.css";
function App() {
  const [data, updateData] = useState({
    username: "",
    phoneNo: ""
  });
  const [tableData, setTableData] = useState([]);

  const changeHandler = (evt) => {
    updateData({ ...data, [evt.target.name]: evt.target.value });
  };
  const submitHandler = (e) => {
    // e.preventDefault();
    console.log(data);
    setTableData([...tableData, data]);
  };
  useEffect(() => {
    console.log(tableData);
  }, [tableData]);
  return (
    <div className="App">
      <form>
        <label>Username:</label>
        <input type="text" name="username" onChange={changeHandler} />
        <label>Phone No:</label>
        <input type="number" name="phoneNo" onChange={changeHandler} />
        <button onClick={submitHandler}>Add</button>
      </form>
    </div>
  );
}

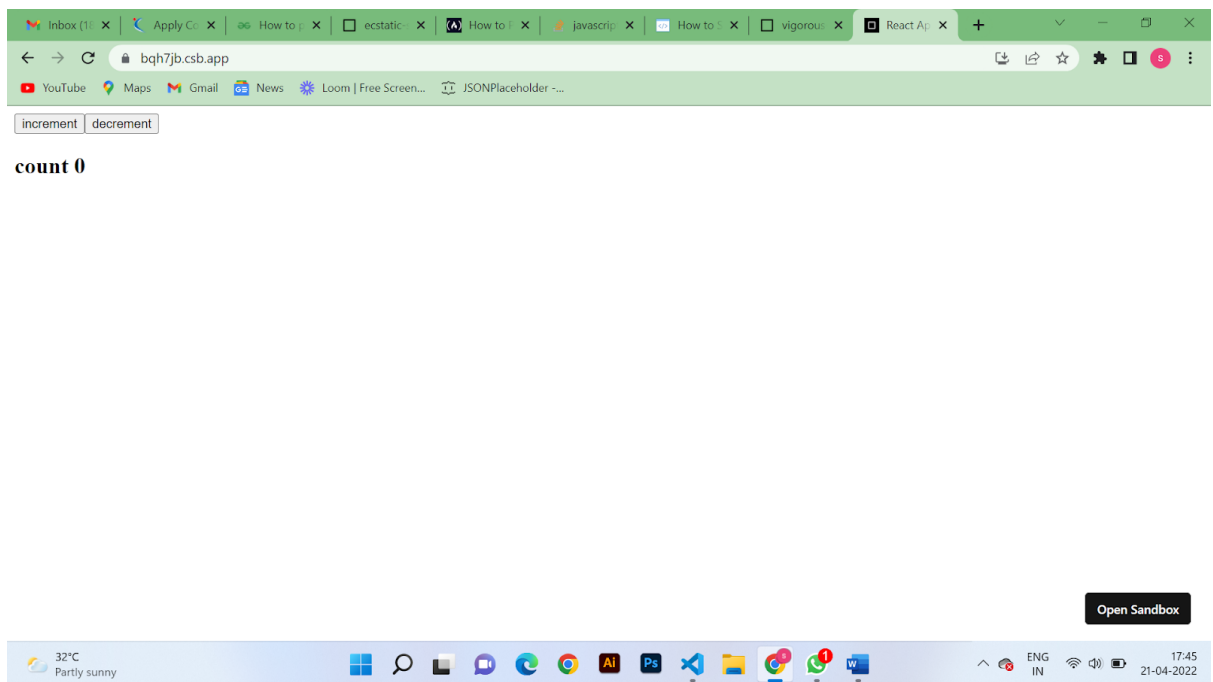
```

```

</from>
{/* {tableData ? (
  ""
) : ( */}
<table className="display">
  <tr>
    <th>Username</th>
    <th>phoneNo</th>
  </tr>
  {tableData.map((item, index) => (
    <tbody key={index}>
      <tr>
        <td>{item.username}</td>
        <td>{item.phoneNo}</td>
      </tr>
    </tbody>
  ))}
</table>
{/* }} */}
</div>
);
}
export default App;

```

Increment & decrement by using Callback hook.



```

import { useCallback, useState } from "react";
export default function App() {

```

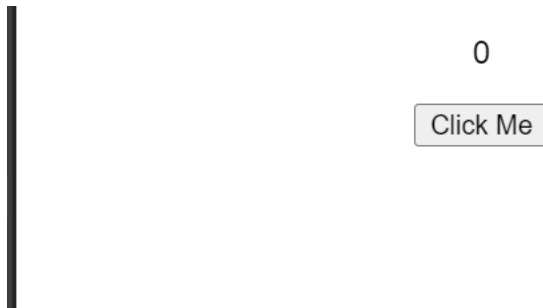


```

const [count, setCount] = useState(0);
const callback = useCallback((count) => {
  setCount(count);}, []);
return (
  <div className="App">
    <Counter parentCallback={callback} />
    <h2>count {count}</h2>
  </div>
)
const Counter = ({ parentCallback }) => {
  const [count, setCount] = useState(0);
  return (
    <div>
      <button onClick={() => { setCount((count) => count + 1);
        parentCallback(count); }}>
        increment
      </button>
      <button onClick={() => {setCount((count) => count - 1);
        parentCallback(count);}}>Decrement
    </button>
  </div>
);
}

```

Data pass from child to parent for increment counter



```

import { useState } from "react";
import Child from "./Child";
import "./styles.css";

export default function App() {
  const [count, setCount] = useState(0);

  const childToParent = () => {
    setCount(count + 1);
  };
  return (

```

```

    <div className="App">
      <p>{count}</p>
      <Child childToParent={childToParent} />
    </div>
  );
}

```

students totalFee,paid,due output in console?

```

const students=[{
  name:"ramu",
  totalFee:10000,
  paid:5000
},
{
  name:"prakash",
  totalFee:25000,
  paid:15000
},
{
  name:"Nani",
  totalFee:30000,
  paid:25000
}];
var totalFee=0;
let paid=0;
var due=0;
let result=students.map ((item)=>{
  console.log(item)
  totalFee+=item.totalFee
  paid+=item.paid
  return(item);
});
due=totalFee-paid
students.map((item)=>{
  return{...students,due}
});console.log(totalFee,due,paid)

```

OUTPUT:

```

    ▶ {name: "ramu", totalFee: 10000, paid: 5000}
    ▶ {name: "prakash", totalFee: 25000, paid: 15000}
    ▶ {name: "Nani", totalFee: 30000, paid: 25000}

65000 20000 45000

```

23. 2 inputs fields data enter print data table

```
import { useEffect, useState } from "react";

export default function Form() {
  const [data, setData] = useState({
    name: "",
    phoneno: ""
  });

  const [listdata, setListData] = useState([]);

  function handleChange(e) {
    setData({ ...data, [e.target.name]: e.target.value });
  }

  const handleClick = (e) => {
    e.preventDefault();
    setListData([...listdata, data]);
  };

  useEffect(() => {
    console.log("data");
  }, [listdata]);

  return (
    <div className="App">
      <form className="formdata">
        <label>Name</label>
        <input
          type="text"
          name="name"
          value={data.name}
          onChange={handleChange}
        />
        <label>Phone No</label>
        <input
          type="number"
          name="phoneno"

```

```

        value={data.phoneno}
        onChange={handleChange}
      />
      <button type="submit" onClick={handlerClick}>
        Add
      </button>
    </form>
    <p>
      <List Listing={listdata} />
    </p>
  </div>
);
}

```

```

const List = ({ Listing }) => {
  return (
    <table className="tabledata" cellPadding="0" cellSpacing="0">
      <tr>
        <th>Name</th>
        <th>Phone No</th>
      </tr>
      <tbody>
        {Listing.map((item) => (
          <tr>
            <td>{item.name}</td>
            <td>{item.phoneno}</td>
          </tr>
        ))}
      </tbody>
    </table>
  );
};

```

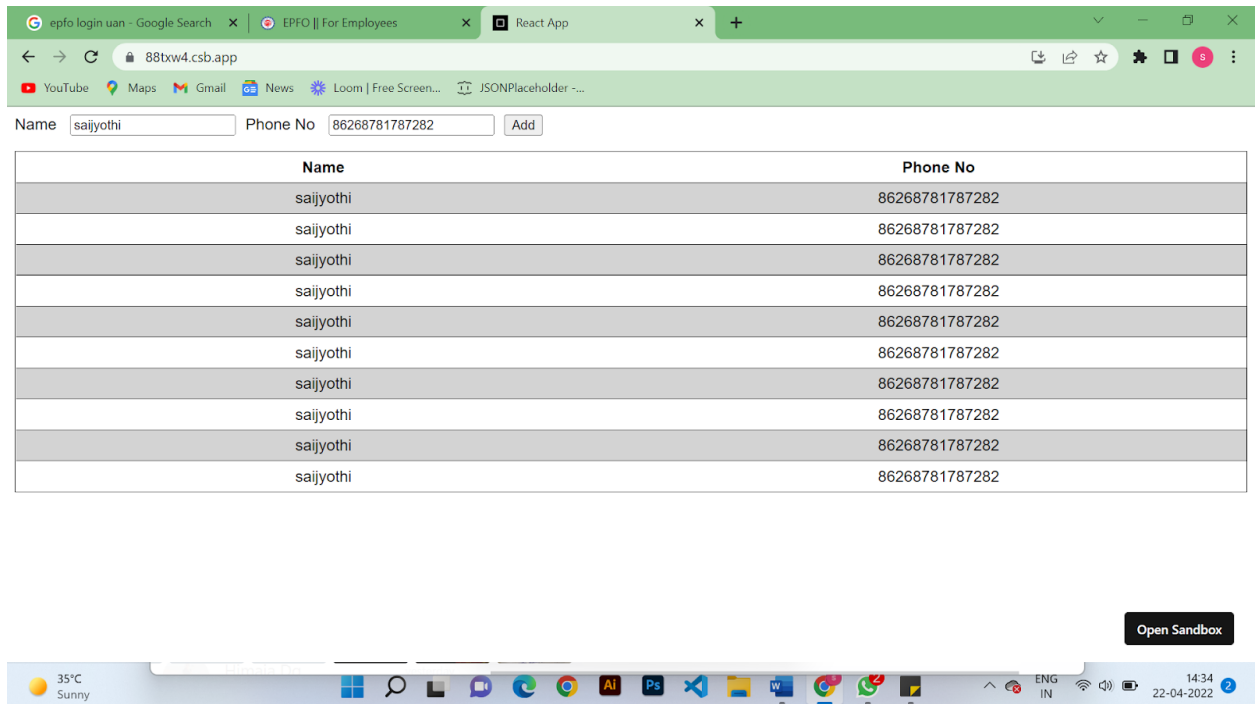
CSS

```

.formdata label {
  padding: 0 14px 0 0;
}
.formdata input {
  margin: 0 10px 0 0;
}
.tabledata {
  width: 100%;
  text-align: center;
  border: 1px solid #000000;
}
.tabledata th {
  padding: 6px 0;
}
.tabledata td {
  border-top: 1px solid #000000;
  padding: 6px 0;
}

```

```
margin: 0;
width: 50%;
}
.tabledata tbody tr:nth-child(odd) {
background-color: lightgray;
}
```



24. Enter value in input field that value show

```
import { useState } from "react";
import "./styles.css";

export default function App() {
  const [count, setCount] = useState(0);
  const [value, setValue] = useState(0);
  const increment = () => {
    if (value) {
      setCount(count + +value);
    }
  };
  const decrement = () => {
    if (value) {
      setCount(count - +value);
    }
  };
  const handlechange = (e) => {
    setValue(e.target.value);
  };
}
```

```

return (
  <div className="App">
    <input type="number" onChange={handlechange} />
    <br />
    <br />
    <button onClick={increment}>Inc</button>
    <span> {count} </span>
    <button onClick={decrement}>Dec</button>
  </div>
);
}

```



25. Increment counter by using a button. pass data from child to parent

App.js

```

import { useState } from "react";
import Child from "./Child";
import "./styles.css";

export default function App() {
  const [count, setCount] = useState(0);

  const childToParent = () => {
    setCount(count + 1);
  };

  return (
    <div className="App">
      <p>{count}</p>
      <Child childToParent={childToParent} />
    </div>
  );
}

```

Child.js

```

const Child = ({ childToParent }) => {
  return (
    <div>
      <button type="button" onClick={() => childToParent()}>
        Click Me
      </button>
    </div>
  );
};

```

```
export default Child;
```

26. increment & decrement counter by using callback

```
import { useCallback, useState } from "react";
```

```
export default function App() {  
  const [count, setCount] = useState(0);  
  
  const callback = useCallback((count) => {  
    setCount(count);  
  }, []);
```

```
  return (  
    <div className="App">  
      <Counter parentCallback={callback} />  
      <h2>count {count}</h2>  
    </div>  
  );  
}
```

```
const Counter = ({ parentCallback }) => {  
  const [count, setCount] = useState(0);
```

```
  return (  
    <div>  
      <button  
        onClick={() => {  
          setCount((count) => count + 1);  
          parentCallback(count);  
        }}  
      >  
        increment  
      </button>  
      <button  
        onClick={() => {  
          setCount((count) => count - 1);  
          parentCallback(count);  
        }}  
      >  
        decrement  
      </button>  
    </div>  
  );  
};
```

27. write a react program with increment ,decrement and auto increment and stop ?

Increment, decrement, automatic, stop counter

```
import { useEffect, useState } from "react";  
import "./styles.css";
```

```

export default function App() {
  //const [count, setCount] = useState(0);
  const [seconds, setSeconds] = useState(0);
  const [isActive, setIsActive] = useState(false);
  const Incrementing = () => {
    setSeconds(seconds + 1);
  };
  const Decrementing = () => {
    if (seconds > 0) setSeconds(seconds - 1);
  };
  const toggle = () => {
    setIsActive(!isActive);
  };
  useEffect(() => {
    let interval = null;
    if (isActive) {
      interval = setInterval(() => {
        setSeconds(seconds + 1);
      }, 1000);
    } else if (!isActive && seconds !== 0) {
      clearInterval(interval);
    }
    return () => clearInterval(interval);
  }, [isActive, seconds]);
  return (
    <div className="App">
      <p> {seconds}</p>
      <button onClick={Incrementing}>Increment</button>
      { /* {seconds} */ }
      <button onClick={Decrementing}>Decrement</button>
    </div>

    <button
      className={` ${isActive ? "active" : "inactive"} `}
      onClick={toggle}
    >
      {isActive ? "stop" : "start"}
    </button>
  </div>
  </div>
  );
}

```

```

//Nearest value print in given array
var counts = [4, 9, 15, 6, 2],
    goal = 20;

```

```

var closest = counts.reduce(function(prev, curr) {
  return (Math.abs(curr - goal) < Math.abs(prev - goal) ? curr : prev);
});

```



```
console.log(closest);
```

28. Fetch Json data and display the json data in the component?

```
import React, { Component } from "react";
import { Container, Grid, Header, List } from "semantic-ui-react";
import contacts from './data.json';
```

```
const App = () => {
```

```
  return (
    <Container>
      <Grid>
        <Grid.Row>
          <Grid.Column>
            <Header>List</Header>
            <List>
              {contacts.map(el => {
                return (
                  <List.Item key={el.id}>
                    <List.Content>
                      {el.firstname} {el.lastname}
                    </List.Content>
                    <List.Content>{el.phone}</List.Content>
                  </List.Item>
                );
              })}
            </List>
          </Grid.Column>
        </Grid.Row>
      </Grid>
    </Container>
  );
}
```

```
export default App;
```

```
Data.json
```

```
[
  {
    "id": 1,
    "firstname": "abc",
    "lastname": "xyz",
    "phone": "+91 789654123",
    "email": "abcyz@gmail.com"
  },
  {
    "id": 2,
```

```

    "firstname": "def",
    "lastname": "uvz",
    "phone": "+91 123456987",
    "email": "defvu@gmail.com"
  }
]

```

29. write a program to display list view and click on any one of the item/list it shows details about the particular list and goto back ?

App.js

```

import React from "react";
import ReactDOM from "react-dom";
import { BrowserRouter as Router, Route } from "react-router-dom";
import List from "./List";
import Detail from "./Detail";
import "./styles.css";

```

```

function App() {
  return (
    <Router className="App">
      <Route exact path="/" component={List} />
      <Route path="/:id" component={Detail} />
    </Router>
  );
}
export default App;

```

List.js

```

import React, { useState } from "react";
import { Link } from "react-router-dom";
function List() {
  // Placeholder for list data
  const [data] = useState([
    { id: "24", title: "Item 1" },
    { id: "25", title: "Item 2" },
    { id: "26", title: "Item 3" },
    { id: "27", title: "Item 4" }
  ]);
  return (
    <div className="container">
      <ul className="list">
        {data.map(item => (
          <li key={item.id} className="list-item">
            <Link to={item.id}>{item.title}</Link>
          </li>
        ))}
      </ul>
    </div>
  );
}
export default List;

```

```

Detail.js
import React, { useState } from "react";
import { Link } from "react-router-dom";
// Placeholder for detail data
function Detail({
  match: {
    params: { id }
  }
}) {
  const [data] = useState({ title: "Item" });
  return (
    <div className="container">
      <div className="card">
        <h1>{data.title}</h1>
        <h3>ID: {id}</h3>
        <Link to="/">Return to List View</Link>
      </div>
    </div>
  );
}

export default Detail;

```

30. pie chart program in React js

```

App.js
import React from "react";
import { data } from "../Data";
import Chart from "../LineChart";

const App = () => {
  return (
    <div>
      <Chart chartData={data} />
    </div>
  );
};
export default App;

LineChart.js
import * as React from "react";
import Highcharts from "highcharts";
import HighchartsReact from "highcharts-react-official";
import HC_exporting from "highcharts/modules/exporting";
HC_exporting(Highcharts);

function Chart(props) {
  const chartOptions = {
    chart: {
      type: "pie",
      height: 250,
      margin: [0, 0, 0, 0]
    }
  };

```

```

    },
    title: {
      verticalAlign: "middle",
      floating: true,
      text: ""
    },
    legend: {
      layout: "vertical",
      align: "right",
      verticalAlign: "top",
      padding: -10,
      symbolRadius: 0,
      itemMarginTop: 5,
      itemMarginBottom: 5,
      symbolPadding: 0,
      symbolWidth: 0,
      symbolHeight: 0,
      squareSymbol: false
    },
    plotOptions: {
      pie: {
        allowPointSelect: true,
        cursor: "pointer",
        dataLabels: {
          enabled: false
        },
        showInLegend: true
      }
    },
    series: [
      {
        data: props.chartData,
        size: "80%",
        innerSize: "50%,"
      }
    ]
  };
  return <HighchartsReact highcharts={Highcharts} options={chartOptions} />;
}
export default Chart;
data.js
export const data = [
  {
    y: 15,
    name: "Category 1"
  },
  {
    y: 25,
    name: "Category 2"
  },
  {

```

```

    y: 35,
    name: "Category 3"
  },
];

```

<https://codesandbox.io/s/increment-and-decrement-in-redux-forked-kkomkj?file=/package.json>

31. Hoc

App.js

```

import React, { Component } from 'react';
import withStorage from './components/withStorage';
class App extends Component {
  state = {
    user: "",
    movie: ""
  }
  loadData() {
    const user = this.props.load('user');
    const movie = this.props.load('movie');
    if (!user || !movie) {
      this.setState({
        user: 'Noel',
        movie: 'Avatar'
      });
    }
    else {
      this.setState({
        user, movie
      });
    }
  }
  componentDidMount() {
    this.loadData()
  }
  handleSubmit = e => {
    e.preventDefault();
    this.props.save('user', this.user.value);
    this.props.save('movie', this.movie.value);
    this.loadData();
    let vm = this;
    setTimeout(() => {
      vm.user.value = "";
      vm.movie.value = "";
    }, 1000)
  }
  render() {
    const { user, movie } = this.state;

```

```

    if (!user || !movie) {
      return <div>Loading...</div>
    }
    return (
      <div className="container">
        <header>
          <h1>React HOC Example</h1>
          <p>Based off this <a
href="https://levelup.gitconnected.com/understanding-react-higher-order-components-by-example-95e8c47c8006" target="_blank">tutorial</a> by <strong>Trey
Huffi</strong> of <a href="https://gitconnected.com"
target="_blank">GitConnected</a></p>
        </header>
        <section>
          <form onSubmit={this.handleSubmit}>
            <div className="form-input">
              <input placeholder="Your name" ref={input => (this.user = input)} />
            </div>
            <div className="form-input">
              <input placeholder="Favorite movie" ref={input => (this.movie = input)} />
            </div>
            <button>Save</button>
          </form>
        </section>
        <section id="result">
          <h3>My name is <strong>{user}</strong> and I liked the movie
<span>{movie}</span></h3>
        </section>
      </div>
    );
  }
}
const HOCComponent = withStorage(App);
export default HOCComponent;

```

[withStorage.js](#)

```

import React, { Component } from "react";
const withStorage = TargetComp => {
  class HOC extends Component {
    state = {
      localStorageAvailable: true
    };
    componentDidMount() {
      this.checkLocalStorageExists();
    }
    checkLocalStorageExists() {
      const testKey = "test";
      try {
        localStorage.setItem(testKey, testKey);
        localStorage.removeItem(testKey);
        this.setState({ localStorageAvailable: true });
      }
    }
  }
}

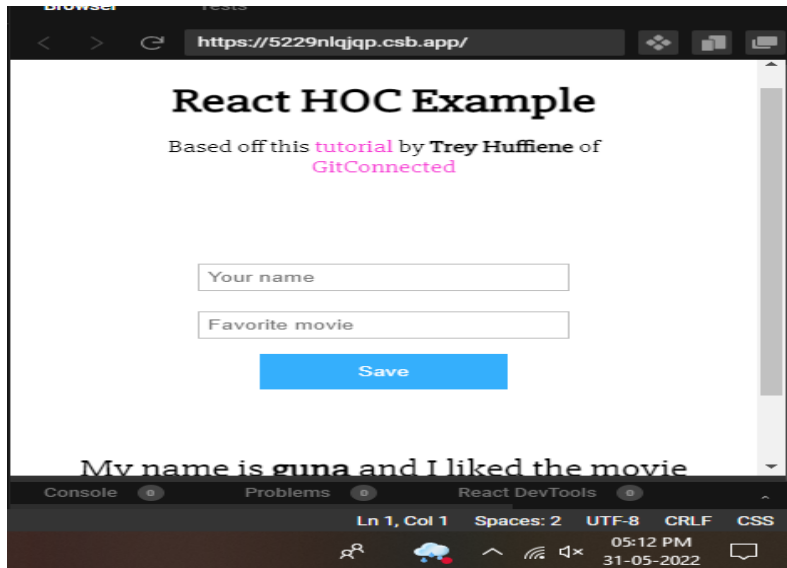
```

```

    } catch (error) {
      this.setState({ localStorageAvailable: false });
    }
  }
  load = key => {
    if (this.state.localStorageAvailable) {
      return localStorage.getItem(key);
    }
    return null;
  };
  save = (key, data) => {
    if (this.state.localStorageAvailable) {
      localStorage.setItem(key, data);
    }
  };
  remove = key => {
    if (this.state.localStorageAvailable) {
      localStorage.removeItem(key);
    }
  };
  render() {
    return (
      <TargetComp
        load={this.load}
        save={this.save}
        remove={this.remove}
        {...this.props}
      />
    );
  }
}
return HOC;
};
export default withStorage;

```

output:



32. Counting How many times render the text field the data in React js

```
import { useEffect, useRef, useState } from "react";
import "./styles.css";
```

```
export default function App() {
  const [data, setData] = useState("");
  const count = useRef(0);
  useEffect(() => {
    count.current = count.current + 1;
  });
  return (
    <div className="App">
      <input
        type="text"
        value={data}
        onChange={(e) => setData(e.target.value)}
      />
      <p>Count:{count.current}</p>
    </div>
  );
}
```

infield

Count:21

```
const { useState } = require("react");
let aa = [];
const App = () => {
  const [name, setName] = useState([]);
  const handelChange = (e) => {
```



```
    console.log(e.target.value);
    aa = [...name, e.target.value];
    setName(aa);
  };
  return (
    <div>
      <div></div>
      <div>
        <input type="text" name="a" onChange={handelChange} />
      </div>
      <div>
        {aa.map((ele, index) => {
          return <p key={index}>{ele}</p>;
        })}
      </div>
    </div>
  );
};
export default App;
output:
G
GU
GUN
GUNA
```