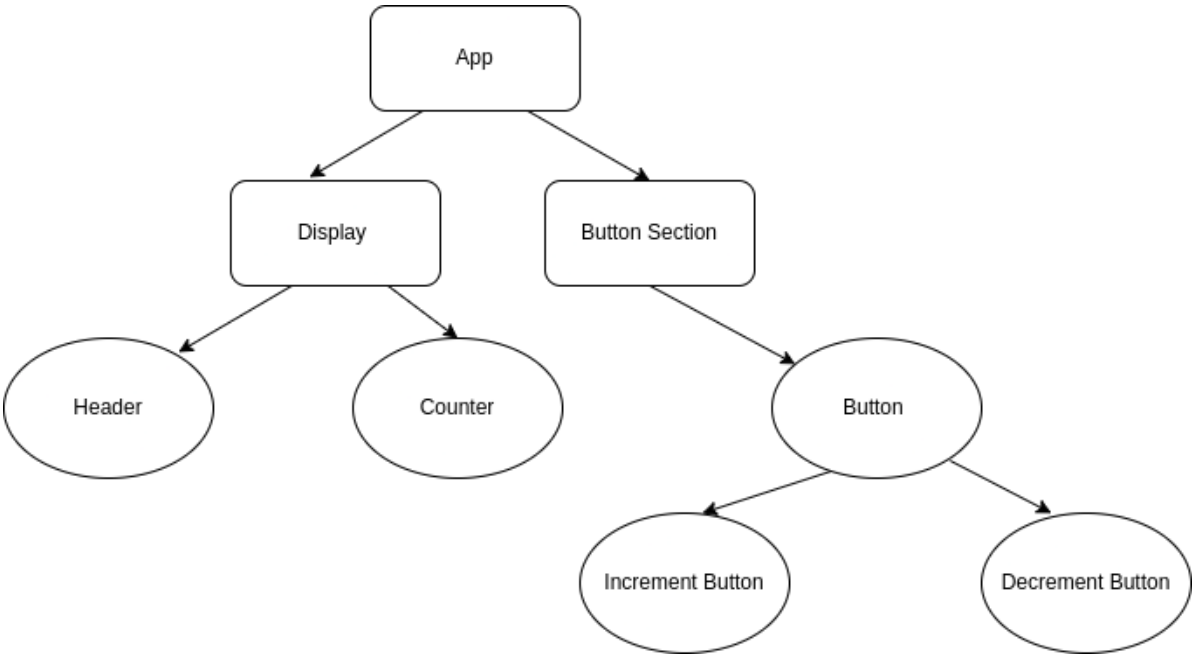
Ex. No: 6	React- Simple Applications

Aim: Implement a react counter app as below.

Mockup:

Component Hierarchy Diagram:



.

Program :

App.jsx:

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

function DisplayBox({count}) {
  return (
    <label type="text" id="display">
      {count}
    </label>
  );
}

function ValButtons({count,setCount}) {
  return (
    <div>
      <button className="inc" onClick={() => setCount(count +
1)}>Increment</button>
      <button className="dec" onClick={() => setCount(count -
1)}>Decrement</button>
      <button className="res" onClick={() => setCount(0)}>Reset</button>
    </div>
  )
}

function Layout() {
  const [count, setCount] = useState(0);
  return (
    <>
      <h1>Counter App</h1>
      <DisplayBox
        count={count}/>
      <ValButtons
        count={count}
        setCount={setCount}/>
    </>
```

```
  );
}

function App() {
  return (
    <Layout />
  );
}

export default App;
```

main.jsx

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'
import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)
```

App.css

```
#root {
  max-width: 1280px;
  margin: 0 auto;
  padding: 2rem;
  text-align: center;
}

button.inc {
  background-color: #199600;
}

button.dec {
  background-color: rgb(184, 10, 10);
}
```

```
button.res {
  background-color: gray;
}

div {
  padding-top: 30px;
}
```

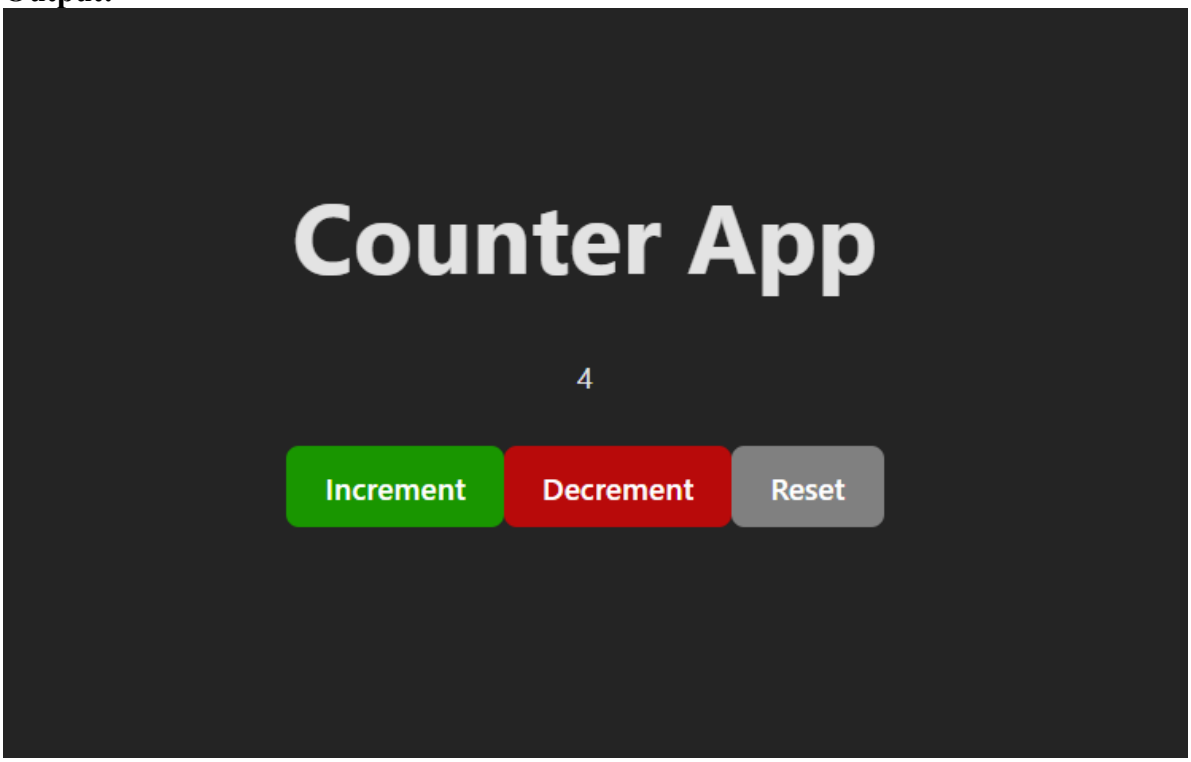
index.css

```
:root {
  font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif;
  line-height: 1.5;
  font-weight: 400;
  font-size:larger;

  color-scheme: light dark;
  color: rgba(255, 255, 255, 0.87);
  background-color: #242424;

  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  -webkit-text-size-adjust: 100%;
}

a {
  font-weight: 500;
  color: #646cff;
  text-decoration: inherit;
}
a:hover {
  color: #535bf2;
}

body {
  margin: 0;
  display: flex;
```

```
  place-items: center;
  min-width: 320px;
  min-height: 100vh;
}

h1 {
  font-size: 3.2em;
  line-height: 1.1;
}

button {
  border-radius: 8px;
  border: 1px solid transparent;
  padding: 0.6em 1.2em;
  font-size: 1em;
  font-weight: 500;
  font-family: inherit;
  background-color: #1a1a1a;
  cursor: pointer;
  transition: border-color 0.25s;
}
button:hover {
  border-color: #646cff;
}
button:focus,
button:focus-visible {
  outline: 4px auto -webkit-focus-ring-color;
}

@media (prefers-color-scheme: light) {
  :root {
    color: #213547;
    background-color: #ffffff;
  }
  a:hover {
    color: #747bff;
  }
  button {
    background-color: #f9f9f9;
```

```
  }
}
```

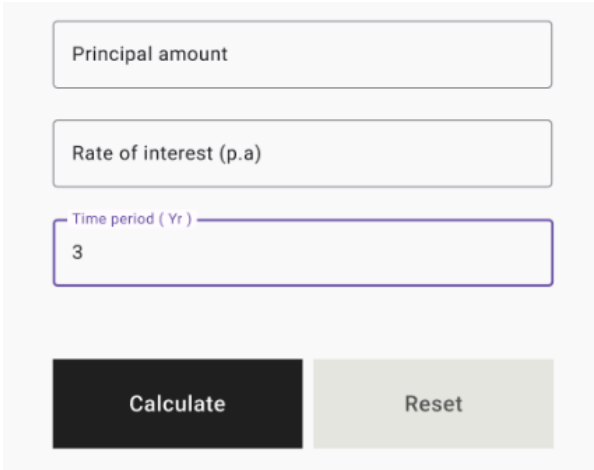
Output:



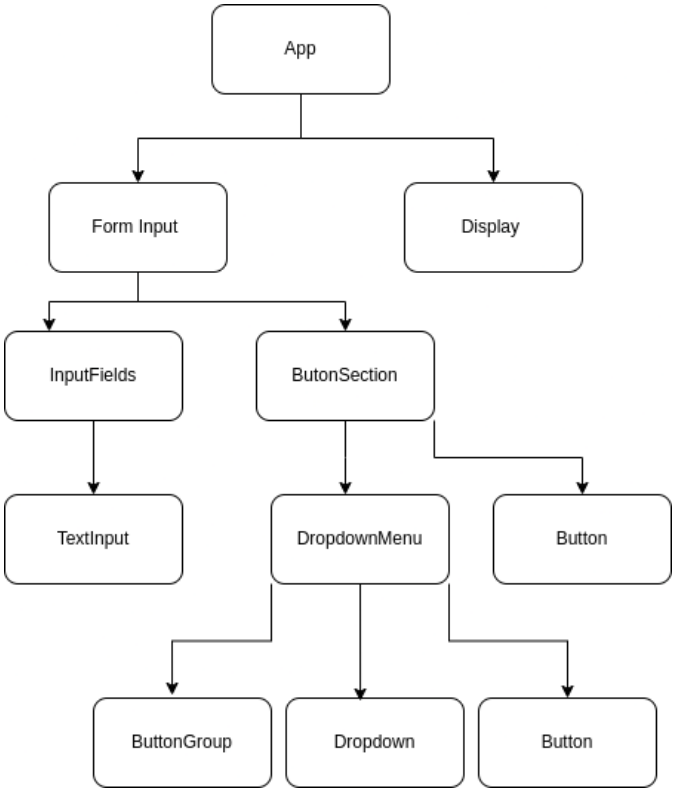
Aim:

Implement a simple and compound interest calculator as below.

Mockup:



Component Hierarchy Diagram:



Program:

App.jsx

```jsx
import React from 'react';
import SICalc from './SICalc';
import CICalc from './CICalc';

function App() {
  return (
    <div>
      <SICalc />
      <CICalc />
    </div>
  );
}

export default App;
```

SICalc.jsx

```jsx
import React, { useState } from 'react';

function SICalc() {
  const [principal, setPrincipal] = useState('');
  const [rate, setRate] = useState('');
  const [time, setTime] = useState('');
  const [result, setResult] = useState('');

  const calculateSimpleInterest = () => {
    const interest = (principal * rate * time) / 100;
    setResult(`Simple Interest: ${interest.toFixed(2)}`);
  };

  return (
    <div>
      <h2>Simple Interest Calculator</h2>
      <div>
        <label>Principal:</label>
        <input type="number" value={principal} onChange={(e) =>
    setPrincipal(e.target.value)} />
      </div>
      <div>
        <label>Rate:</label>
        <input type="number" value={rate} onChange={(e) =>
    setRate(e.target.value)} />
      </div>
      <div>
        <label>Time (years):</label>
        <input type="number" value={time} onChange={(e) =>
    setTime(e.target.value)} />
      </div>
      <button onClick={calculateSimpleInterest}>Calculate</button>
      <div>{result}</div>
    </div>
  );
}
export default SICalc;
```

CICalc.jsx

```jsx
import React, { useState } from 'react';

function CICalc() {
  const [principal, setPrincipal] = useState('');
  const [rate, setRate] = useState('');
  const [time, setTime] = useState('');
  const [result, setResult] = useState('');
  const calculateCompoundInterest = () => {
    const compoundInterest = principal * Math.pow(1 + rate / 100, time) -
    principal;
    setResult(`Compound Interest: ${compoundInterest.toFixed(2)}`);
  };
  return (
    <div>
      <h2>Compound Interest Calculator</h2>
      <div>
        <label>Principal:</label>
        <input type="number" value={principal} onChange={(e) =>
    setPrincipal(e.target.value)} />
      </div>
      <div>
        <label>Rate:</label>
        <input type="number" value={rate} onChange={(e) =>
    setRate(e.target.value)} />
      </div>
      <div>
        <label>Time (years):</label>
        <input type="number" value={time} onChange={(e) =>
    setTime(e.target.value)} />
      </div>
      <button onClick={calculateCompoundInterest}>Calculate</button>
      <div>{result}</div>
    </div>
  );
}
export default CICalc;
```

main.jsx

```jsx
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'
import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)
```

App.css

```css
#root {
  max-width: 1280px;
  margin: 0 auto;
  padding: 2rem;
  text-align: center;
}
```

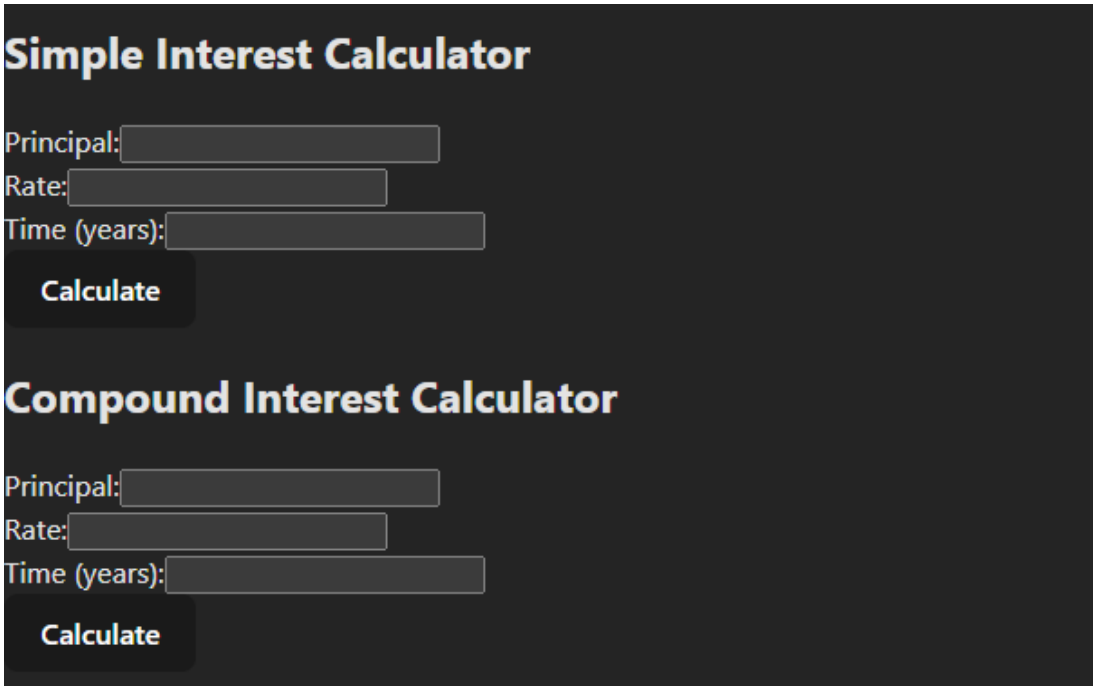
index.css

```css
:root {
  font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif;
  line-height: 1.5;
  font-weight: 400;

  color-scheme: light dark;
  color: rgba(255, 255, 255, 0.87);
  background-color: #242424;

  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  -webkit-text-size-adjust: 100%;
}

a {
```

```
  font-weight: 500;
  color: #646cff;
  text-decoration: inherit;
}
a:hover {
  color: #535bf2;
}

body {
  margin: 0;
  display: flex;
  place-items: center;
  min-width: 320px;
  min-height: 100vh;
}

h1 {
  font-size: 3.2em;
  line-height: 1.1;
}

button {
  border-radius: 8px;
  border: 1px solid transparent;
  padding: 0.6em 1.2em;
  font-size: 1em;
  font-weight: 500;
  font-family: inherit;
  background-color: #1a1a1a;
  cursor: pointer;
  transition: border-color 0.25s;
}
button:hover {
  border-color: #646cff;
}
button:focus,
button:focus-visible {
  outline: 4px auto -webkit-focus-ring-color;
}
```

```css
@media (prefers-color-scheme: light) {
  :root {
    color: #213547;
    background-color: #ffffff;
  }
  a:hover {
    color: #747bff;
  }
  button {
    background-color: #f9f9f9;
  }
}
```

Output:



Result:

Thus, simple react applications were written and studied.