ReactJS-HOL (Mandatory Hands On)

Create a React Application named “cricketapp”

Code:

*// src/EvenPlayers.js*

*import* React *from* 'react'**;**

const EvenPlayers **=** ({ **team** }) **=>** {

  const players **=** team**.**filter((**\_,** **index**) **=>** index **%** 2 **!==** 0)**;**

  const ordinals **=** ['Second'**,** 'Fourth'**,** 'Sixth'**,** 'Eighth']**;**

**return** (

    <ul>

      {players**.**map((**player,** **index**) **=>** (

        <li *key***=**{index}>

          {ordinals[index] **||** `Player ${index **\*** 2 **+** 2}`} : {player}

        </li>

      ))}

    </ul>

  )**;**

}**;**

*export* **default** EvenPlayers**;**

*// src/ListofIndianPlayers.js*

*import* React *from* 'react'**;**

const ListofIndianPlayers **=** ({ **IndianPlayers** }) **=>** {

**return** (

    <ul>

      {IndianPlayers**.**map((**player,** **index**) **=>** (

        <li *key***=**{index}>{player}</li>

      ))}

    </ul>

  )**;**

}**;**

*export* **default** ListofIndianPlayers**;**

*// src/ListofPlayers.js*

*import* React *from* 'react'**;**

const ListofPlayers **=** ({ **players** }) **=>** {

**return** (

    <ul>

      {players**.**map((**item,** **index**) **=>** (

        <li *key***=**{index}>

          Mr. {item**.***name*} - <span>{item**.***score*}</span>

        </li>

      ))}

    </ul>

  )**;**

}**;**

*export* **default** ListofPlayers**;**

*// src/OddPlayers.js*

*import* React *from* 'react'**;**

const OddPlayers **=** ({ **team** }) **=>** {

  const players **=** team**.**filter((**\_,** **index**) **=>** index **%** 2 **!==** 1)**;**

  const ordinals **=** ['First'**,** 'Third'**,** 'Fifth'**,** 'Seventh']**;**

**return** (

    <ul>

      {players**.**map((**player,** **index**) **=>** (

        <li *key***=**{index}>

          {ordinals[index] **||** `Player ${index **\*** 2 **+** 1}`} : {player}

        </li>

      ))}

    </ul>

  )**;**

}**;**

*export* **default** OddPlayers**;**

*// src/Scorebelow70.js*

*import* React *from* 'react'**;**

const Scorebelow70 **=** ({ **players** }) **=>** {

  const below70 **=** players**.**filter(**item** **=>** item**.***score* **<=** 70)**;**

**return** (

    <ul>

      {below70**.**map((**item,** **index**) **=>** (

        <li *key***=**{index}>

          Mr. {item**.***name*} - <span>{item**.***score*}</span>

        </li>

      ))}

    </ul>

  )**;**

}**;**

*export* **default** Scorebelow70**;**

*// src/App.js*

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

*import* './App.css'**;**

*import* ListofPlayers *from* './Components/ListofPlayers'**;**

*import* Scorebelow70 *from* './Components/Scorebelow70'**;**

*import* OddPlayers *from* './Components/OddPlayers'**;**

*import* EvenPlayers *from* './Components/EvenPlayers'**;**

*import* ListofIndianPlayers *from* './Components/ListofIndianPlayers'**;**

*import* { players**,** IndianTeam**,** IndianPlayers } *from* './data'**;**

function App() {

  const [flag**,** setFlag] **=** useState(true)**;**

**return** (

    <div *className***=**"App">

      <button *onClick***=**{() **=>** setFlag(**!**flag)}>

        {flag **?** "Show Indian Team View" **:** "Show Player Score View"}

      </button>

      {flag **?** (

        <div>

          <h1>List of Players</h1>

          <ListofPlayers *players***=**{players} />

          <hr />

          <h1>List of Players having Scores Less than 70</h1>

          <Scorebelow70 *players***=**{players} />

        </div>

      ) **:** (

        <div>

          <h1>Indian Team</h1>

          <h1>Odd Players</h1>

          <OddPlayers *team***=**{IndianTeam} />

          <hr />

          <h1>Even Players</h1>

          <EvenPlayers *team***=**{IndianTeam} />

          <hr />

          <h1>List of Indian Players Merged:</h1>

          <ListofIndianPlayers *IndianPlayers***=**{IndianPlayers} />

        </div>

      )}

    </div>

  )**;**

}

*export* **default** App**;**

*export* const players **=** [

  { name**:** 'Rohit'**,** score**:** 95 }**,**

  { name**:** 'Virat'**,** score**:** 88 }**,**

  { name**:** 'Rahul'**,** score**:** 62 }**,**

  { name**:** 'Pant'**,** score**:** 45 }**,**

  { name**:** 'Hardik'**,** score**:** 75 }**,**

  { name**:** 'Jadeja'**,** score**:** 68 }**,**

  { name**:** 'Ashwin'**,** score**:** 72 }**,**

  { name**:** 'Shami'**,** score**:** 40 }**,**

  { name**:** 'Bumrah'**,** score**:** 85 }**,**

  { name**:** 'Surya'**,** score**:** 65 }**,**

  { name**:** 'Gill'**,** score**:** 90 }

]**;**

*export* const IndianTeam **=** [

  'Sachin1'**,** 'Dhoni2'**,** 'Virat3'**,** 'Rohit4'**,** 'Yuvaraj5'**,** 'Raina6'

]**;**

*export* const IndianPlayers **=** [

  'Mr. First Player'**,**

  'Mr. Second Player'**,**

  'Mr. Third Player'**,**

  'Mr. Fourth Player'**,**

  'Mr. Fifth Player'**,**

  'Mr. Sixth Player'

]**;**

body {

  margin**:** 0**;**

  font-family**:** *-apple-system***,** BlinkMacSystemFont**,** 'Segoe UI'**,** 'Roboto'**,** 'Oxygen'**,**

    'Ubuntu'**,** 'Cantarell'**,** 'Fira Sans'**,** 'Droid Sans'**,** 'Helvetica Neue'**,**

    sans-serif**;**

*-webkit-font-smoothing***:** antialiased**;**

*-moz-osx-font-smoothing***:** grayscale**;**

}

code {

  font-family**:** source-code-pro**,** Menlo**,** Monaco**,** Consolas**,** 'Courier New'**,**

    monospace**;**

}

*import* React *from* 'react'**;**

*import* ReactDOM *from* 'react-dom/client'**;**

*import* './index.css'**;**

*import* App *from* './App'**;**

*import* reportWebVitals *from* './reportWebVitals'**;**

const root **=** ReactDOM**.**createRoot(document**.**getElementById('root'))**;**

root**.**render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

)**;**

*// If you want to start measuring performance in your app, pass a function*

*// to log results (for example: reportWebVitals(console.log))*

*// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals*

reportWebVitals()**;**

const reportWebVitals **=** **onPerfEntry** **=>** {

**if** (onPerfEntry **&&** onPerfEntry **instanceof** Function) {

**import**('web-vitals')**.**then(({ **getCLS,** **getFID,** **getFCP,** **getLCP,** **getTTFB** }) **=>** {

      getCLS(onPerfEntry)**;**

      getFID(onPerfEntry)**;**

      getFCP(onPerfEntry)**;**

      getLCP(onPerfEntry)**;**

      getTTFB(onPerfEntry)**;**

    })**;**

  }

}**;**

*export* **default** reportWebVitals**;**

<!DOCTYPE *html*>

<html *lang***=**"en">

  <head>

    <meta *charset***=**"utf-8" />

    <link *rel***=**"icon" *href***=**"%PUBLIC\_URL%/favicon.ico" />

    <meta *name***=**"viewport" *content***=**"width=device-width, initial-scale=1" />

    <meta *name***=**"theme-color" *content***=**"#000000" />

    <meta

*name***=**"description"

*content***=**"Web site created using create-react-app"

    />

    <link *rel***=**"apple-touch-icon" *href***=**"%PUBLIC\_URL%/logo192.png" />

*<!--*

*manifest.json provides metadata used when your web app is installed on a*

*user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/*

*-->*

    <link *rel***=**"manifest" *href***=**"%PUBLIC\_URL%/manifest.json" />

*<!--*

*Notice the use of %PUBLIC\_URL% in the tags above.*

*It will be replaced with the URL of the `public` folder during the build.*

*Only files inside the `public` folder can be referenced from the HTML.*

*Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will*

*work correctly both with client-side routing and a non-root public URL.*

*Learn how to configure a non-root public URL by running `npm run build`.*

*-->*

    <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div ***id*=**"root"></div>

*<!--*

*This HTML file is a template.*

*If you open it directly in the browser, you will see an empty page.*

*You can add webfonts, meta tags, or analytics to this file.*

*The build step will place the bundled scripts into the <body> tag.*

*To begin the development, run `npm start` or `yarn start`.*

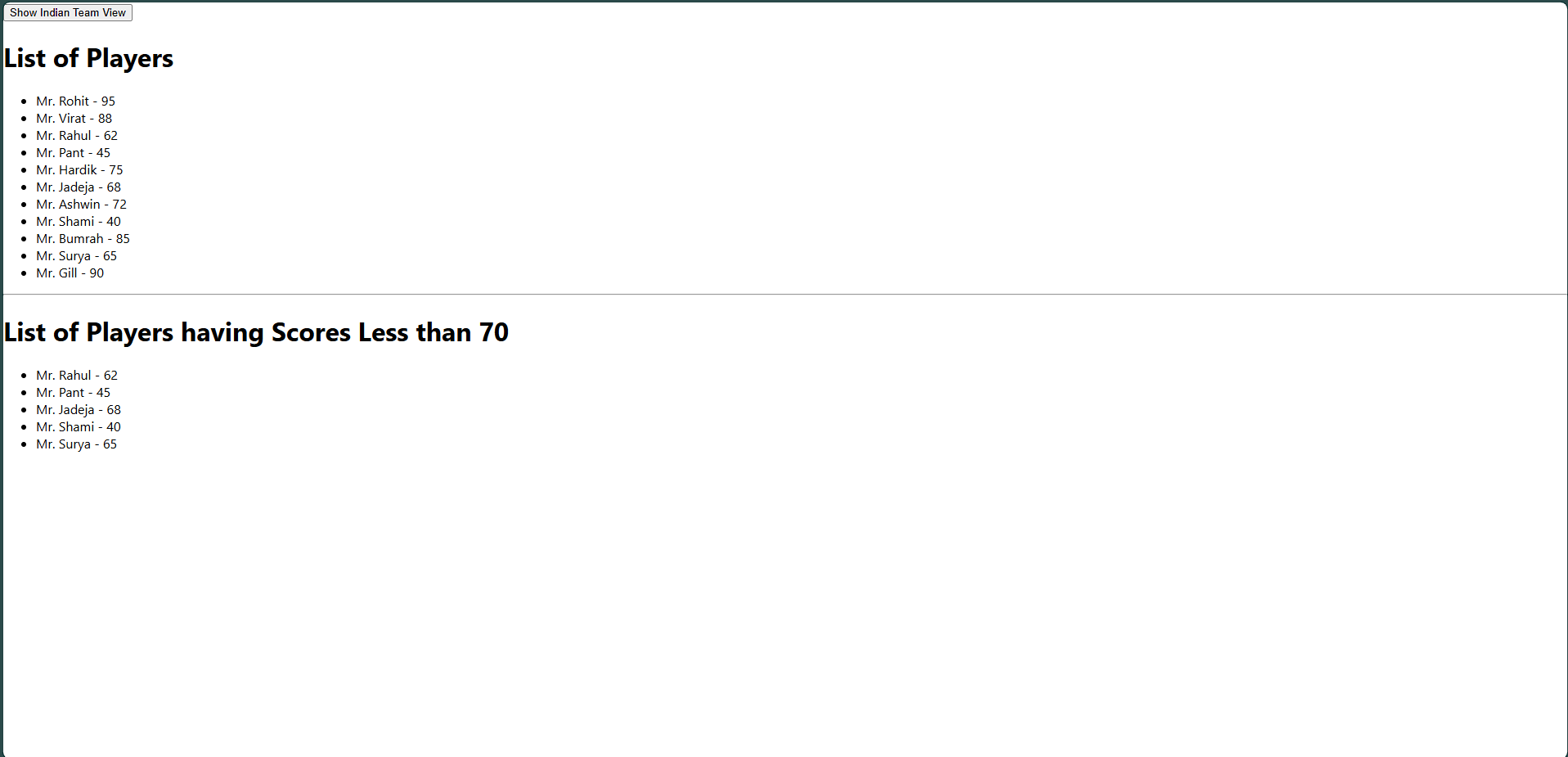
*To create a production bundle, use `npm run build` or `yarn build`.*

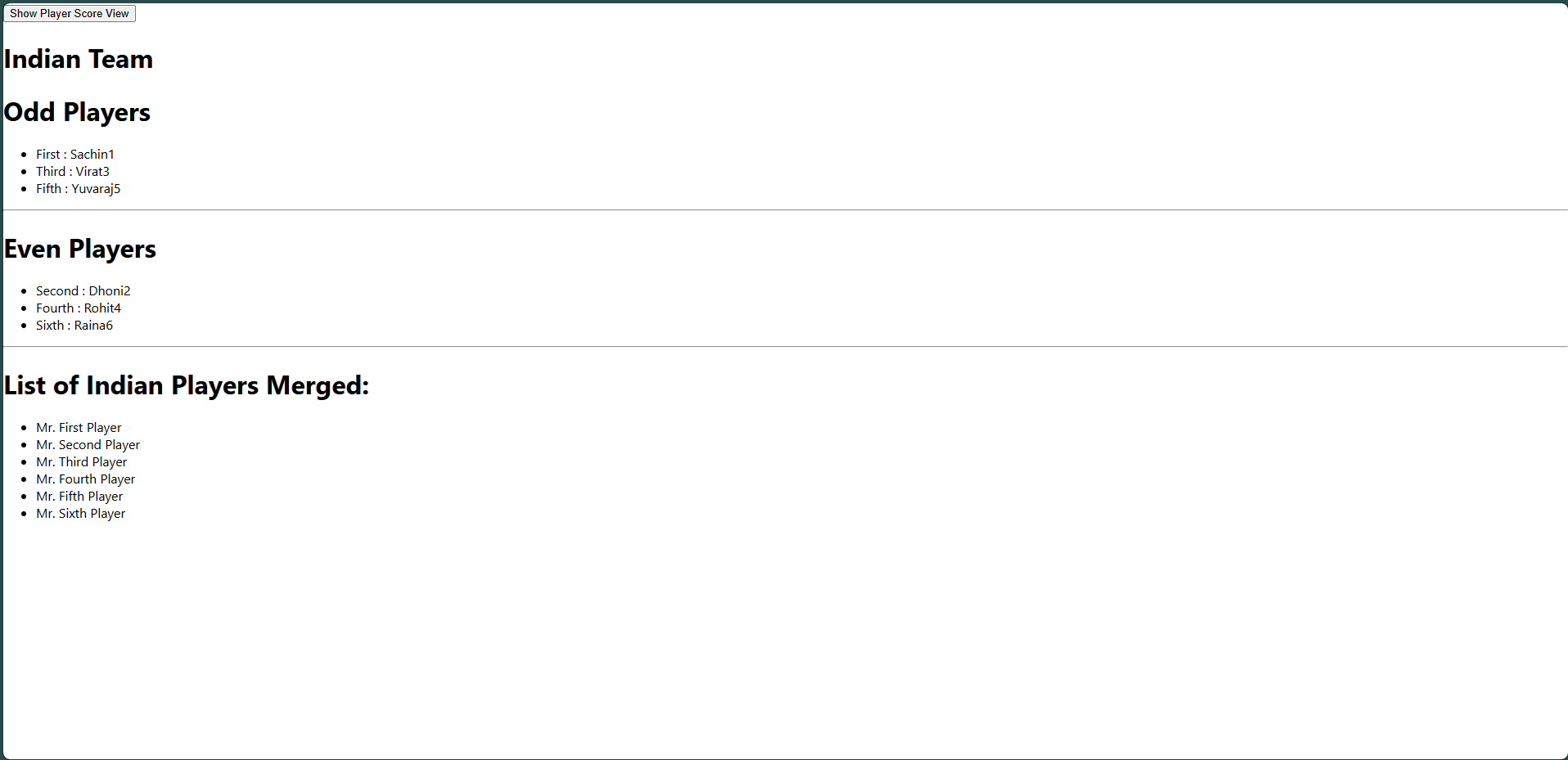
*-->*

  </body>

</html>

Output:





Create a React Application named “officespacerentalapp”

Code:

<!DOCTYPE *html*>

<html *lang***=**"en">

  <head>

    <meta *charset***=**"utf-8" />

    <link *rel***=**"icon" *href***=**"%PUBLIC\_URL%/favicon.ico" />

    <meta *name***=**"viewport" *content***=**"width=device-width, initial-scale=1" />

    <meta *name***=**"theme-color" *content***=**"#000000" />

    <meta

*name***=**"description"

*content***=**"Web site created using create-react-app"

    />

    <link *rel***=**"apple-touch-icon" *href***=**"%PUBLIC\_URL%/logo192.png" />

*<!--*

*manifest.json provides metadata used when your web app is installed on a*

*user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/*

*-->*

    <link *rel***=**"manifest" *href***=**"%PUBLIC\_URL%/manifest.json" />

*<!--*

*Notice the use of %PUBLIC\_URL% in the tags above.*

*It will be replaced with the URL of the `public` folder during the build.*

*Only files inside the `public` folder can be referenced from the HTML.*

*Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will*

*work correctly both with client-side routing and a non-root public URL.*

*Learn how to configure a non-root public URL by running `npm run build`.*

*-->*

    <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div ***id*=**"root"></div>

*<!--*

*This HTML file is a template.*

*If you open it directly in the browser, you will see an empty page.*

*You can add webfonts, meta tags, or analytics to this file.*

*The build step will place the bundled scripts into the <body> tag.*

*To begin the development, run `npm start` or `yarn start`.*

*To create a production bundle, use `npm run build` or `yarn build`.*

*-->*

  </body>

</html>

*/\* src/App.css \*/*

*.App* {

  text-align**:** center**;**

  padding**:** 20px**;**

  font-family**:** Arial**;**

*/\* background: red; \*/*

  display**:** flex**;**

  align-items**:** center**;**

  justify-content**:** center**;**

  flex-wrap**:** wrap**;**

}

*.card* {

  border**:** 1px solid **#ccc;**

  margin**:** 20px auto**;**

  padding**:** 15px**;**

  max-width**:** 400px**;**

  box-shadow**:** 0 0 10px rgba(0**,** 0**,** 0**,** 0.1)**;**

}

*.textRed* {

  color**:** red**;**

}

*.textGreen* {

  color**:** green**;**

}

h1{

  text-align**:** center **!important;**

  width**:** 100%**;**

}

*// src/App.js*

*import* React *from* 'react'**;**

*import* './App.css'**;**

*import* { offices } *from* './data'**;**

*import* OfficeCard *from* './OfficeCard'**;**

function App() {

**return** (

    <>

      <h1>Office Space at Affordable Range</h1>

      <div *className***=**"App">

        {offices**.**map((**office,** **index**) **=>** (

          <OfficeCard *key***=**{index} *office***=**{office} />

        ))}

      </div>

    </>

  )**;**

}

*export* **default** App**;**

*export* const offices **=** [

  { Name**:** "DBS"**,** Rent**:** 50000**,** Address**:** "Chennai"**,** Image**:** "https://images.unsplash.com/photo-1706689656095-168768dc20a5" }**,**

  { Name**:** "Regus"**,** Rent**:** 75000**,** Address**:** "Mumbai"**,** Image**:** "https://images.unsplash.com/photo-1555474488-d2282fe0593f" }**,**

  { Name**:** "WeWork"**,** Rent**:** 45000**,** Address**:** "Bangalore"**,** Image**:** "https://images.unsplash.com/photo-1549399905-5d1bad747576" }**,**

  { Name**:** "SmartSpace"**,** Rent**:** 85000**,** Address**:** "Delhi"**,** Image**:** "https://images.unsplash.com/photo-1705909773420-8d7af2a343f9" }**,**

]**;**

*// src/OfficeCard.js*

*import* React *from* 'react'**;**

*import* './App.css'**;**

const OfficeCard **=** ({ **office** }) **=>** {

  const rentClass **=** office**.***Rent* **<=** 60000 **?** 'textRed' **:** 'textGreen'**;**

**return** (

    <div *className***=**"card">

      <img *src***=**{office**.***Image*} *width***=**"25%" *height***=**"25%" *alt***=**"Office Space" />

      <h3>Name: {office**.***Name*}</h3>

      <h3 *className***=**{rentClass}>Rent: Rs. {office**.***Rent*}</h3>

      <h3>Address: {office**.***Address*}</h3>

    </div>

  )**;**

}**;**

*export* **default** OfficeCard**;**

const reportWebVitals **=** **onPerfEntry** **=>** {

**if** (onPerfEntry **&&** onPerfEntry **instanceof** Function) {

**import**('web-vitals')**.**then(({ **getCLS,** **getFID,** **getFCP,** **getLCP,** **getTTFB** }) **=>** {

      getCLS(onPerfEntry)**;**

      getFID(onPerfEntry)**;**

      getFCP(onPerfEntry)**;**

      getLCP(onPerfEntry)**;**

      getTTFB(onPerfEntry)**;**

    })**;**

  }

}**;**

*export* **default** reportWebVitals**;**

*// jest-dom adds custom jest matchers for asserting on DOM nodes.*

*// allows you to do things like:*

*// expect(element).toHaveTextContent(/react/i)*

*// learn more: https://github.com/testing-library/jest-dom*

*import* '@testing-library/jest-dom'**;**

Output:



Create a React Application “eventexamplesapp”

Code:

<!DOCTYPE *html*>

<html *lang***=**"en">

  <head>

    <meta *charset***=**"utf-8" />

    <link *rel***=**"icon" *href***=**"%PUBLIC\_URL%/favicon.ico" />

    <meta *name***=**"viewport" *content***=**"width=device-width, initial-scale=1" />

    <meta *name***=**"theme-color" *content***=**"#000000" />

    <meta

*name***=**"description"

*content***=**"Web site created using create-react-app"

    />

    <link *rel***=**"apple-touch-icon" *href***=**"%PUBLIC\_URL%/logo192.png" />

*<!--*

*manifest.json provides metadata used when your web app is installed on a*

*user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/*

*-->*

    <link *rel***=**"manifest" *href***=**"%PUBLIC\_URL%/manifest.json" />

*<!--*

*Notice the use of %PUBLIC\_URL% in the tags above.*

*It will be replaced with the URL of the `public` folder during the build.*

*Only files inside the `public` folder can be referenced from the HTML.*

*Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will*

*work correctly both with client-side routing and a non-root public URL.*

*Learn how to configure a non-root public URL by running `npm run build`.*

*-->*

    <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div ***id*=**"root"></div>

*<!--*

*This HTML file is a template.*

*If you open it directly in the browser, you will see an empty page.*

*You can add webfonts, meta tags, or analytics to this file.*

*The build step will place the bundled scripts into the <body> tag.*

*To begin the development, run `npm start` or `yarn start`.*

*To create a production bundle, use `npm run build` or `yarn build`.*

*-->*

  </body>

</html>

button{

  margin**:** 3px 0**;**

  font-size**:** 1.2rem**;**

  padding**:** 0 20px**;**

}

span{

  font-size**:** 2rem**;**

}

h1{

  font-weight**:** bolder**;**

  font-family**:** Arial**,** Helvetica**,** sans-serif**;**

  font-size**:** 3rem**;**

  color**:** green**;**

}

table tr td{

  font-size**:** 1.5rem **!important;**

}

table tr td input**,** table tr td textarea**,** table tr td button{

  font-size**:** 1.5rem **!important;**

}

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

*import* './App.css'**;**

*import* CurrencyConvertor *from* './CurrencyConvertor'**;**

const App **=** () **=>** {

  const [value**,** setValue] **=** useState(0)**;**

  function increase() {

    setValue(value**+**1)**;**

    alert("Hello! Member!")**;**

**return;**

  }

  function decrease() {

    setValue(value**-**1)**;**

**return;**

  }

  function welcome(**message**) {

    alert(message)**;**

**return;**

  }

  function clickonme() {

    alert("I was clicked !")**;**

**return;**

  }

**return** (

    <div>

      <span>{value}</span>

      <br />

      <button *onClick***=**{increase}>Increment</button>

      <br />

      <button *onClick***=**{decrease}>Decrement</button>

      <br />

      <button *onClick***=**{()**=>**{

        welcome('welcome')

      }}>Say Welcome</button>

      <br />

      <button *onClick***=**{clickonme}>Click on Me</button>

      <br />

      <CurrencyConvertor></CurrencyConvertor>

    </div>

  )

}

*export* **default** App

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

const CurrencyConvertor **=** () **=>** {

    const [amount**,** setAmount] **=** useState(80)**;**

    const [curr**,** setCurr] **=** useState('Euro')**;**

**return** (

        <div>

            <h1>Currency Convertor</h1>

            <table>

                <tr>

                    <td>Amount:</td>

                    <td>

                        <input *type***=**'number' *value***=**{amount} *onInput***=**{(**e**)**=>**{

                            setAmount(e**.***target***.***value*)**;**

                        }} />

                    </td>

                </tr>

                <tr>

                    <td>Currency:</td>

                    <td>

                        <textarea *value***=**{curr} *onInput***=**{(**e**)**=>**{setCurr(e**.***target***.***value*)}}></textarea>

                    </td>

                </tr>

                <tr>

                    <td *colSpan***=**{2} *style***=**{{textAlign**:** 'center'}}>

                        <button *onClick***=**{()**=>**{

                            alert(`Converting to ${curr}. Amount is ${amount**\***80}`)**;**

                        }}>Submit</button>

                    </td>

                </tr>

            </table>

        </div>

    )

}

*export* **default** CurrencyConvertor

*import* React *from* 'react'**;**

*import* ReactDOM *from* 'react-dom/client'**;**

*import* './index.css'**;**

*import* App *from* './App'**;**

*import* reportWebVitals *from* './reportWebVitals'**;**

const root **=** ReactDOM**.**createRoot(document**.**getElementById('root'))**;**

root**.**render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

)**;**

*// If you want to start measuring performance in your app, pass a function*

*// to log results (for example: reportWebVitals(console.log))*

*// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals*

reportWebVitals()**;**

const reportWebVitals **=** **onPerfEntry** **=>** {

**if** (onPerfEntry **&&** onPerfEntry **instanceof** Function) {

**import**('web-vitals')**.**then(({ **getCLS,** **getFID,** **getFCP,** **getLCP,** **getTTFB** }) **=>** {

      getCLS(onPerfEntry)**;**

      getFID(onPerfEntry)**;**

      getFCP(onPerfEntry)**;**

      getLCP(onPerfEntry)**;**

      getTTFB(onPerfEntry)**;**

    })**;**

  }

}**;**

*export* **default** reportWebVitals**;**

*// jest-dom adds custom jest matchers for asserting on DOM nodes.*

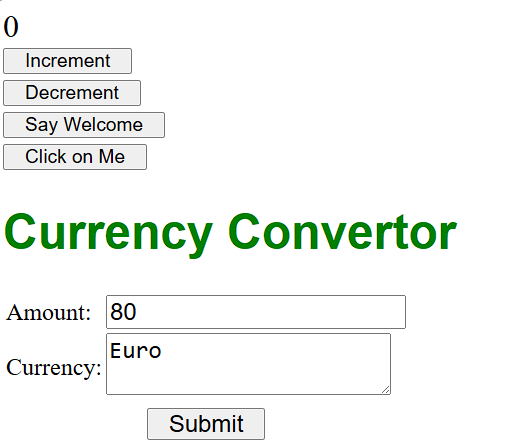
*// allows you to do things like:*

*// expect(element).toHaveTextContent(/react/i)*

*// learn more: https://github.com/testing-library/jest-dom*

*import* '@testing-library/jest-dom'**;**

Output:



Create a React Application named “ticketbookingapp”

Code:

<!DOCTYPE *html*>

<html *lang***=**"en">

  <head>

    <meta *charset***=**"utf-8" />

    <link *rel***=**"icon" *href***=**"%PUBLIC\_URL%/favicon.ico" />

    <meta *name***=**"viewport" *content***=**"width=device-width, initial-scale=1" />

    <meta *name***=**"theme-color" *content***=**"#000000" />

    <meta

*name***=**"description"

*content***=**"Web site created using create-react-app"

    />

    <link *rel***=**"apple-touch-icon" *href***=**"%PUBLIC\_URL%/logo192.png" />

*<!--*

*manifest.json provides metadata used when your web app is installed on a*

*user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/*

*-->*

    <link *rel***=**"manifest" *href***=**"%PUBLIC\_URL%/manifest.json" />

*<!--*

*Notice the use of %PUBLIC\_URL% in the tags above.*

*It will be replaced with the URL of the `public` folder during the build.*

*Only files inside the `public` folder can be referenced from the HTML.*

*Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will*

*work correctly both with client-side routing and a non-root public URL.*

*Learn how to configure a non-root public URL by running `npm run build`.*

*-->*

    <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div ***id*=**"root"></div>

*<!--*

*This HTML file is a template.*

*If you open it directly in the browser, you will see an empty page.*

*You can add webfonts, meta tags, or analytics to this file.*

*The build step will place the bundled scripts into the <body> tag.*

*To begin the development, run `npm start` or `yarn start`.*

*To create a production bundle, use `npm run build` or `yarn build`.*

*-->*

  </body>

</html>

h1 {

  font-size**:** 24px**;**

}

button {

  margin-top**:** 10px**;**

  padding**:** 5px 10px**;**

}

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

*import* './App.css'**;**

function LoginButton({ **onClick** }) {

**return** <button *onClick***=**{onClick}>Login</button>**;**

}

function LogoutButton({ **onClick** }) {

**return** <button *onClick***=**{onClick}>Logout</button>**;**

}

function GuestGreeting() {

**return** (

    <div>

      <h1>Please sign up.</h1>

      <FlightList />

    </div>

  )**;**

}

function UserGreeting() {

**return** (

    <div>

      <h1>Welcome back</h1>

      <BookTicket />

    </div>

  )**;**

}

function Greeting({ **isLoggedIn** }) {

**return** isLoggedIn **?** <UserGreeting /> **:** <GuestGreeting />**;**

}

*// Example guest view*

function FlightList() {

**return** (

    <ul>

      <li>Flight: AI202 - Delhi to Mumbai</li>

      <li>Flight: AI305 - Bangalore to Chennai</li>

    </ul>

  )**;**

}

*// Example user view*

function BookTicket() {

**return** (

    <div>

      <h2>Book Your Flight:</h2>

      <p>Flight: AI202</p>

      <button *onClick***=**{()**=>**{

        alert("TICKET BOOKED")**;**

      }}>Confirm Booking</button>

    </div>

  )**;**

}

function App() {

  const [isLoggedIn**,** setIsLoggedIn] **=** useState(false)**;**

  const handleLoginClick **=** () **=>** setIsLoggedIn(true)**;**

  const handleLogoutClick **=** () **=>** setIsLoggedIn(false)**;**

  const button **=** isLoggedIn **?** (

    <LogoutButton *onClick***=**{handleLogoutClick} />

  ) **:** (

    <LoginButton *onClick***=**{handleLoginClick} />

  )**;**

**return** (

    <div>

      <Greeting *isLoggedIn***=**{isLoggedIn} />

      {button}

    </div>

  )**;**

}

*export* **default** App**;**

body {

  margin**:** 0**;**

  font-family**:** *-apple-system***,** BlinkMacSystemFont**,** 'Segoe UI'**,** 'Roboto'**,** 'Oxygen'**,**

    'Ubuntu'**,** 'Cantarell'**,** 'Fira Sans'**,** 'Droid Sans'**,** 'Helvetica Neue'**,**

    sans-serif**;**

*-webkit-font-smoothing***:** antialiased**;**

*-moz-osx-font-smoothing***:** grayscale**;**

}

code {

  font-family**:** source-code-pro**,** Menlo**,** Monaco**,** Consolas**,** 'Courier New'**,**

    monospace**;**

}

*import* React *from* 'react'**;**

*import* ReactDOM *from* 'react-dom/client'**;**

*import* './index.css'**;**

*import* App *from* './App'**;**

*import* reportWebVitals *from* './reportWebVitals'**;**

const root **=** ReactDOM**.**createRoot(document**.**getElementById('root'))**;**

root**.**render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

)**;**

*// If you want to start measuring performance in your app, pass a function*

*// to log results (for example: reportWebVitals(console.log))*

*// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals*

reportWebVitals()**;**

const reportWebVitals **=** **onPerfEntry** **=>** {

**if** (onPerfEntry **&&** onPerfEntry **instanceof** Function) {

**import**('web-vitals')**.**then(({ **getCLS,** **getFID,** **getFCP,** **getLCP,** **getTTFB** }) **=>** {

      getCLS(onPerfEntry)**;**

      getFID(onPerfEntry)**;**

      getFCP(onPerfEntry)**;**

      getLCP(onPerfEntry)**;**

      getTTFB(onPerfEntry)**;**

    })**;**

  }

}**;**

*export* **default** reportWebVitals**;**

*// jest-dom adds custom jest matchers for asserting on DOM nodes.*

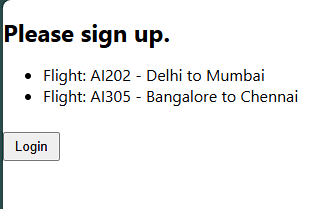
*// allows you to do things like:*

*// expect(element).toHaveTextContent(/react/i)*

*// learn more: https://github.com/testing-library/jest-dom*

*import* '@testing-library/jest-dom'**;**

Output:



Create a React App named “bloggerapp”

Code:

<!DOCTYPE *html*>

<html *lang***=**"en">

  <head>

    <meta *charset***=**"utf-8" />

    <link *rel***=**"icon" *href***=**"%PUBLIC\_URL%/favicon.ico" />

    <meta *name***=**"viewport" *content***=**"width=device-width, initial-scale=1" />

    <meta *name***=**"theme-color" *content***=**"#000000" />

    <meta

*name***=**"description"

*content***=**"Web site created using create-react-app"

    />

    <link *rel***=**"apple-touch-icon" *href***=**"%PUBLIC\_URL%/logo192.png" />

*<!--*

*manifest.json provides metadata used when your web app is installed on a*

*user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/*

*-->*

    <link *rel***=**"manifest" *href***=**"%PUBLIC\_URL%/manifest.json" />

*<!--*

*Notice the use of %PUBLIC\_URL% in the tags above.*

*It will be replaced with the URL of the `public` folder during the build.*

*Only files inside the `public` folder can be referenced from the HTML.*

*Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will*

*work correctly both with client-side routing and a non-root public URL.*

*Learn how to configure a non-root public URL by running `npm run build`.*

*-->*

    <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div ***id*=**"root"></div>

*<!--*

*This HTML file is a template.*

*If you open it directly in the browser, you will see an empty page.*

*You can add webfonts, meta tags, or analytics to this file.*

*The build step will place the bundled scripts into the <body> tag.*

*To begin the development, run `npm start` or `yarn start`.*

*To create a production bundle, use `npm run build` or `yarn build`.*

*-->*

  </body>

</html>

*.container* {

  display**:** flex**;**

  justify-content**:** space-around**;**

  padding**:** 40px**;**

}

*.section* {

  flex**:** 1**;**

  padding**:** 20px**;**

  border-left**:** 4px solid green**;**

}

*.section:first-child* {

  border-left**:** none**;**

}

h2 {

  font-size**:** 1.5rem**;**

  margin-bottom**:** 1rem**;**

}

p {

  margin**:** 0.5rem 0**;**

}

*import* React *from* 'react'**;**

*import* BookDetails *from* './BookDetails'**;**

*import* BlogDetails *from* './BlogDetails'**;**

*import* CourseDetails *from* './CourseDetails'**;**

*import* './App.css'**;**

function App() {

**return** (

    <div *className***=**"container">

      <CourseDetails />

      <BookDetails />

      <BlogDetails />

    </div>

  )**;**

}

*export* **default** App**;**

function BlogDetails() {

    const blogs **=** [

        {

            id**:** 1**,**

            title**:** "React Learning"**,**

            author**:** "Stephen Biz"**,**

            content**:** "Welcome to learning React!"**,**

        }**,**

        {

            id**:** 2**,**

            title**:** "Installation"**,**

            author**:** "Schwezdenier"**,**

            content**:** "You can install React from npm."**,**

        }**,**

    ]**;**

**return** (

        <div *className***=**"section">

            <h2>Blog Details</h2>

            {blogs**.**map((**blog**) **=>** (

                <div *key***=**{blog**.***id*}>

                    <strong>{blog**.***title*}</strong>

                    <p><b>{blog**.***author*}</b></p>

                    <p>{blog**.***content*}</p>

                </div>

            ))}

        </div>

    )**;**

}

*export* **default** BlogDetails**;**

function BookDetails() {

  const books **=** [

    { id**:** 1**,** bname**:** "Master React"**,** price**:** 670 }**,**

    { id**:** 2**,** bname**:** "Deep Dive into Angular 11"**,** price**:** 800 }**,**

    { id**:** 3**,** bname**:** "Mongo Essentials"**,** price**:** 450 }**,**

  ]**;**

**return** (

    <div *className***=**"section">

      <h2>Book Details</h2>

      {books**.**map((**book**) **=>** (

        <div *key***=**{book**.***id*}>

          <strong>{book**.***bname*}</strong>

          <p>{book**.***price*}</p>

        </div>

      ))}

    </div>

  )**;**

}

*export* **default** BookDetails**;**

function CourseDetails() {

  const courses **=** [

    { name**:** "Angular"**,** date**:** "4/5/2021" }**,**

    { name**:** "React"**,** date**:** "6/3/20201" }**,**

  ]**;**

**return** (

    <div *className***=**"section">

      <h2>Course Details</h2>

      {courses**.**map((**course,** **index**) **=>** (

        <div *key***=**{index}>

          <strong>{course**.***name*}</strong>

          <p>{course**.***date*}</p>

        </div>

      ))}

    </div>

  )**;**

}

*export* **default** CourseDetails**;**

body {

  margin**:** 0**;**

  font-family**:** *-apple-system***,** BlinkMacSystemFont**,** 'Segoe UI'**,** 'Roboto'**,** 'Oxygen'**,**

    'Ubuntu'**,** 'Cantarell'**,** 'Fira Sans'**,** 'Droid Sans'**,** 'Helvetica Neue'**,**

    sans-serif**;**

*-webkit-font-smoothing***:** antialiased**;**

*-moz-osx-font-smoothing***:** grayscale**;**

}

code {

  font-family**:** source-code-pro**,** Menlo**,** Monaco**,** Consolas**,** 'Courier New'**,**

    monospace**;**

}

*import* React *from* 'react'**;**

*import* ReactDOM *from* 'react-dom/client'**;**

*import* './index.css'**;**

*import* App *from* './App'**;**

*import* reportWebVitals *from* './reportWebVitals'**;**

const root **=** ReactDOM**.**createRoot(document**.**getElementById('root'))**;**

root**.**render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

)**;**

*// If you want to start measuring performance in your app, pass a function*

*// to log results (for example: reportWebVitals(console.log))*

*// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals*

reportWebVitals()**;**

Output:

