**9. React Application – cricketapp**

**Objective:**

Build a react application cricketapp with two components i.e.,

* ListofPlayers
* IndianPlayers

**Software Requirements:**

* Node.js and npm
* Visual Studio Code
* Web browser

Steps for creating React Application:

**Step1**: Create a new React App

Initially, open the command prompt and execute the below command to create new react application.

i.e.,

|  |
| --- |
| npx create-react-app cricketapp  cd cricketapp |

Now, again run the below command to redirect the project directory to the VSCode .

|  |
| --- |
| code . |

**Step2:** Implement the ListofPlayers component.

In this,

An array of 11 players with names and scores consists.

Also, the component filters the players with scores below 70 using arrow functions.

i.e., ListofPlayers.js

|  |
| --- |
| import React from "react";  function ListofPlayers() {  // Step 1: Declare an array of players with scores  const players = [  {name: "Rohit Sharma", score: 80},  {name: "Virat Kohli", score: 95},  {name: "KL Rahul", score: 65},  {name: "Shreyas Iyer", score: 40},  { name: "Hardik Pandya", score: 85 },  { name: "Ravindra Jadeja", score: 55 },  { name: "Rishabh Pant", score: 72 },  { name: "Suryakumar Yadav", score: 88 },  { name: "Mohammed Shami", score: 35 },  { name: "Jasprit Bumrah", score: 78 },  { name: "Kuldeep Yadav", score: 60 }  ];  const lowScorers = players.filter (player => player.score < 70);  return (  <div>  <h2>All Players</h2>  <ul>  {players.map ((p, index) => (  <li key={index}>  {p.name} - {p.score}  </li>  ))}  </ul>  <h2>Players with score below 70</h2>  <ul>  {lowScorers.map((p, index) => (  <li key={index}>  {p.name} - {p.score}  </li>  ))}  </ul>  </div>  );  }  export default ListofPlayers; |

**Step3**: Implement IndianPlayers component.

This component will:

Display odd and even index players using destructuring.

Also, merge two arrays using spread operator.

i.e., IndianPlayers.js

|  |
| --- |
| import React from "react";  import "./App.css";  function IndianPlayers () {    const team = ["Sachin1", "Dhoni2", "Virat3", "Rohit4", "Yuvraj5", "Raina6"];    const oddPlayers = team.filter ((\_, index) => index % 2 === 0);    const evenPlayers = team.filter((\_, index) => index % 2 !== 0);    const T20players = [      "Mr. First Player",      "Mr. Second Player",      "Mr. Third Player"    ];    const RanjiPlayers = [      "Mr. Fourth Player",      "Mr. Fifth Player",      "Mr. Sixth Player"    ];    const mergedPlayers = [...T20players, ...RanjiPlayers];    return (      <div>        <h2>Odd Players</h2>        <ul>          <li>First : {oddPlayers[0]}</li>          <li>Third : {oddPlayers[1]}</li>          <li>Fifth : {oddPlayers[2]}</li>        </ul>        <h2>Even Players</h2>        <ul>          <li>Second : {evenPlayers[0]}</li>          <li>Fourth : {evenPlayers[1]}</li>          <li>Sixth : {evenPlayers[2]}</li>        </ul>        <h2>List of Indian Players Merged:</h2>        <ul>          {mergedPlayers.map((p, index) => (            <li key={index}>{p}</li>          ))}        </ul>      </div>    );  }  export default IndianPlayers; |

**Step4**: Modify App.js

In this the outputs are resulted based on flag value.

i.e., App.js

|  |
| --- |
| import React from "react";  import "./App.css";  import ListofPlayers from "./ListofPlayers";  import IndianPlayers from "./IndianPlayers";  function App() {  const flag = true; // Change this to false to see IndianPlayers  return (  <div className="App">  <h1>Cricket App</h1>  {flag ? <ListofPlayers /> : <IndianPlayers />}  </div>  );  }  export default App; |

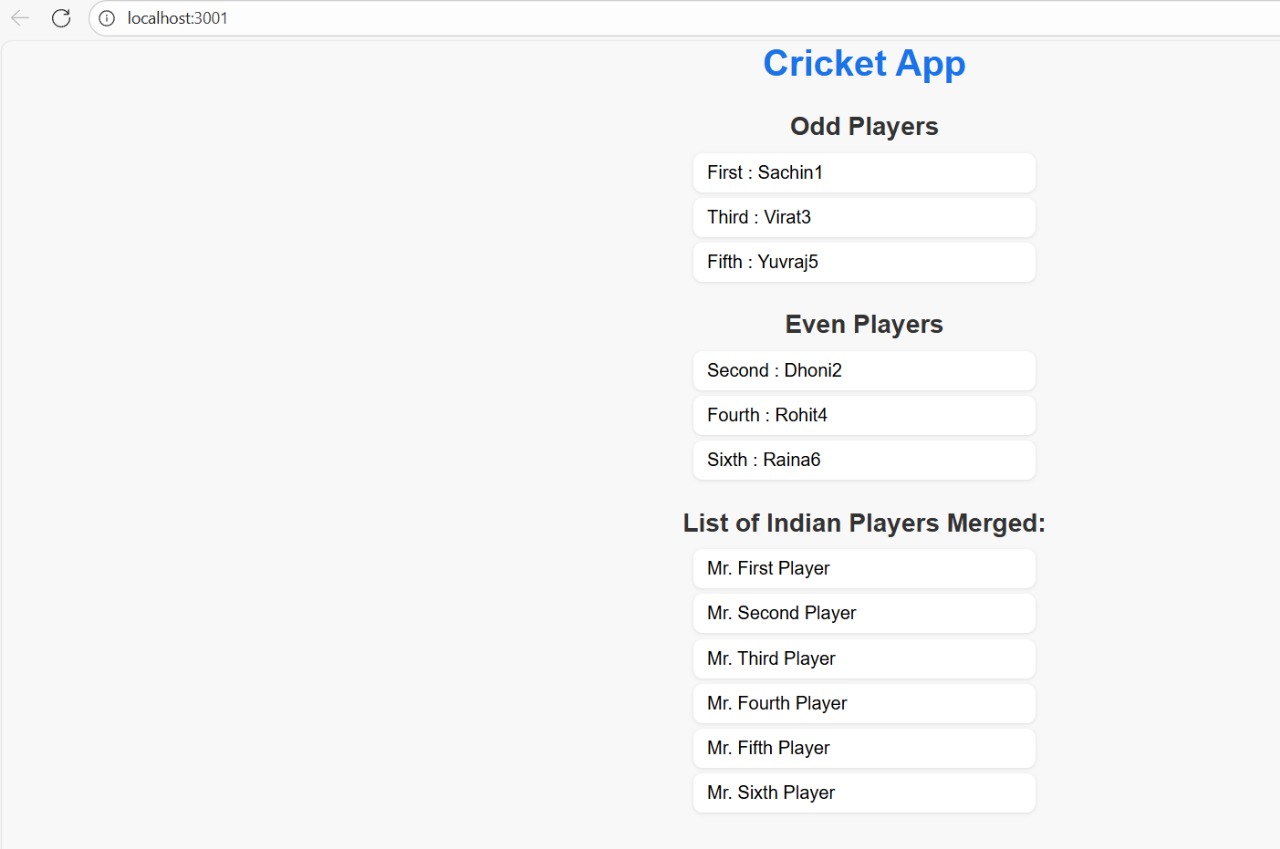
**Step5**: Run the Application.

Open the terminal in VSCode and execute the below command then the development server starts automatically.

|  |
| --- |
| npm start |

**Expected outcome:**

When the flag is false:



When the flag is true:

