## **Objectives**

* List the features of ES6
* Explain JavaScript let
* Identify the differences between var and let
* Explain JavaScript const
* Explain ES6 class fundamentals
* Explain ES6 class inheritance
* Define ES6 arrow functions
* Identify set(), map()

In this hands-on lab, you will learn how to:

* Use map() method of ES6
* Apply arrow functions of ES6
* Implement Destructuring features of ES6

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

Create a React Application named “cricketapp” with the following components:

1. ListofPlayers

* Declare an array with 11 players and store details of their names and scores using the map feature of ES6



* Filter the players with scores below 70 using arrow functions of ES6.



1. IndianPlayers
   1. Display the Odd Team Player and Even Team players using the Destructuring features of ES6



* 1. Declare two arrays T20players and RanjiTrophy players and merge the two arrays and display them using the Merge feature of ES6



Display these two components in the same home page using a simple if else in the flag variable.

**src/components/ListofPlayers.js**

import React from "react";

/\*

Demonstrates:

- create players array using map()

- display with map()

- filter players with arrow functions

\*/

const playerNames = [

"Rohit Sharma", "Virat Kohli", "KL Rahul", "Rishabh Pant",

"Jasprit Bumrah", "Ravindra Jadeja", "Hardik Pandya",

"Shikhar Dhawan", "Yuzvendra Chahal", "Shubman Gill", "Axar Patel"

];

const playerScores = [85, 65, 75, 90, 45, 50, 72, 33, 69, 81, 54];

// create an array of objects using map()

const players = playerNames.map((name, i) => ({ name, score: playerScores[i] }));

const ListofPlayers = () => {

// filter using arrow function

const lowScorers = players.filter(p => p.score < 70);

return (

<div>

<h2>All Players</h2>

<ul>

{players.map((p, idx) => (

<li key={idx}>{p.name} — {p.score}</li>

))}

</ul>

<h3>Players with score &lt; 70 (filtered)</h3>

<ul>

{lowScorers.map((p, idx) => <li key={idx}>{p.name} — {p.score}</li>)}

</ul>

</div>

);

};

export default ListofPlayers;

**src/components/IndianPlayers.js**

import React from "react";

/\*

Demonstrates:

- destructuring to get odd/even teams

- merging two arrays using spread operator

\*/

const players = [

"Player1", "Player2", "Player3", "Player4", "Player5",

"Player6", "Player7", "Player8", "Player9", "Player10", "Player11"

];

const splitOddEven = (arr) => {

const odd = arr.filter((\_, i) => (i + 1) % 2 === 1); // 1-based odd positions

const even = arr.filter((\_, i) => (i + 1) % 2 === 0); // 1-based even positions

return [odd, even]; // return array to be destructured

};

const IndianPlayers = () => {

// destructuring assignment

const [oddTeam, evenTeam] = splitOddEven(players);

const T20players = ["T20-A", "T20-B", "T20-C"];

const RanjiPlayers = ["Ranji-A", "Ranji-B"];

const merged = [...T20players, ...RanjiPlayers]; // merge using ES6 spread

return (

<div>

<h2>Odd Team Players</h2>

<ul>{oddTeam.map((p, i) => <li key={i}>{p}</li>)}</ul>

<h2>Even Team Players</h2>

<ul>{evenTeam.map((p, i) => <li key={i}>{p}</li>)}</ul>

<h2>Merged T20 + Ranji Players</h2>

<ul>{merged.map((p, i) => <li key={i}>{p}</li>)}</ul>

</div>

);

};

export default IndianPlayers;

**src/App.js**

import React from "react";

import ListofPlayers from "./components/ListofPlayers";

import IndianPlayers from "./components/IndianPlayers";

function App() {

const flag = true; // <-- set true to show ListofPlayers, false to show IndianPlayers

return (

<div className="App" style={{padding:20}}>

<h1>Cricket App (React ES6 Features)</h1>

{flag ? <ListofPlayers /> : <IndianPlayers />}

</div>

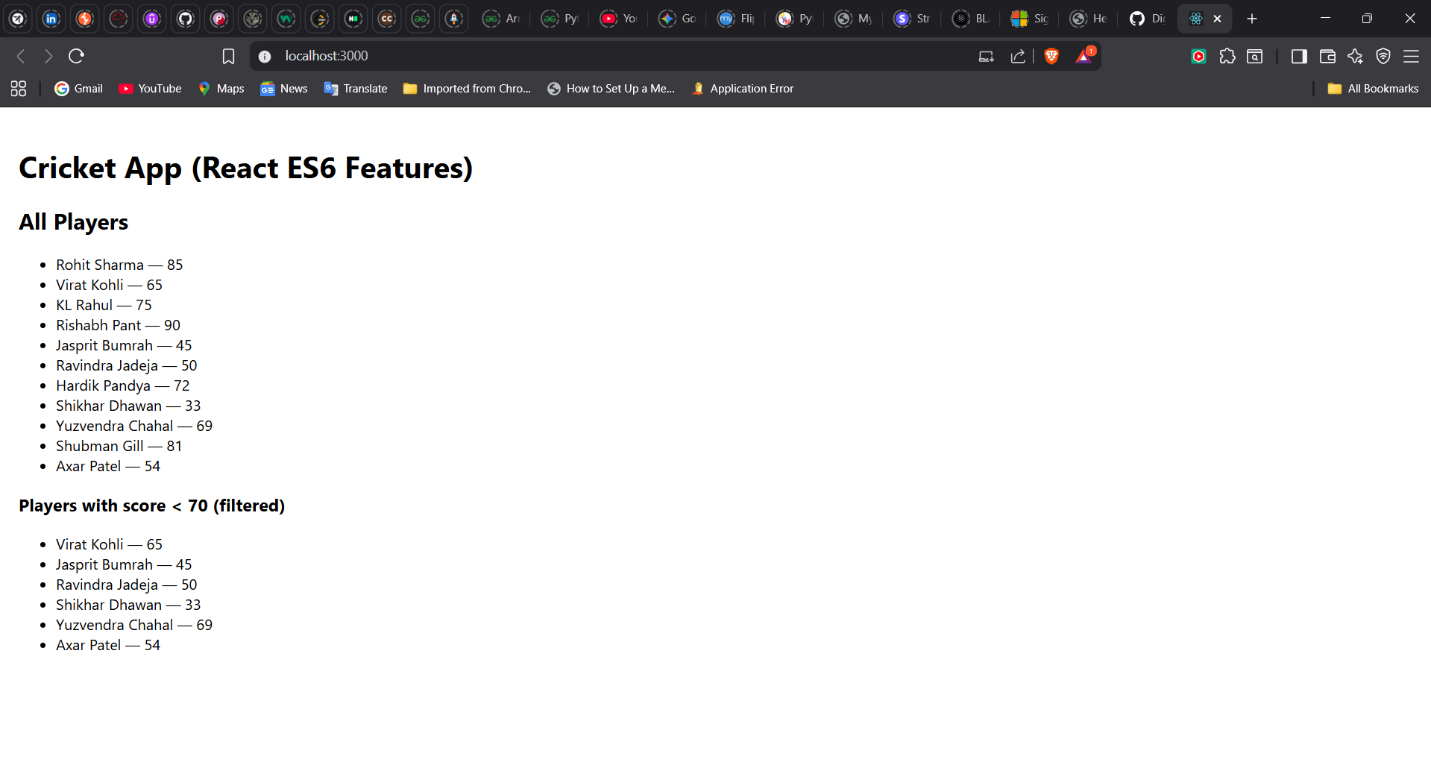
);

}

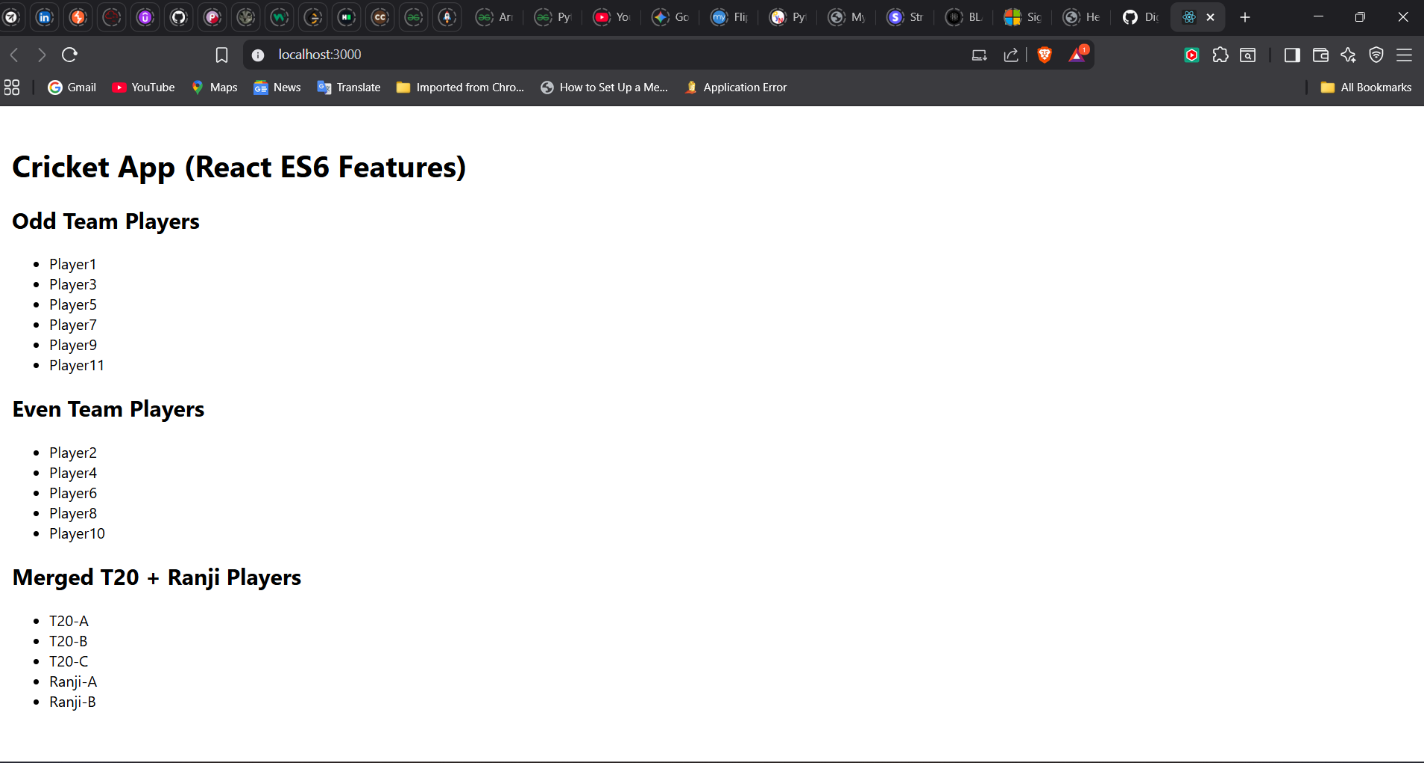
export default App;

**Output:**

When Flag=true



When Flag=false



**Hint:**

