**CricketApp**

**REACT**

Vaishnavi

29/7/2025

ES6 (ECMAScript 2015) introduced significant improvements to JavaScript, making it more powerful and developer-friendly. Features like let, const, arrow functions, destructuring, classes, map(), Set(), and Map() provide cleaner and more efficient coding practices.

This lab demonstrates **React development using ES6 features** by building a **Cricket Application** named cricketapp. The app displays:

A list of cricket players with their scores using map().

Filters players with scores below 70 using arrow functions.

Displays **odd and even team players** using **destructuring**.

Merges two arrays (T20 and Ranji Trophy players) using **spread operator.**

**Objective:**

· **List ES6 Features** such as let, const, arrow functions, destructuring, classes.

· Understand **difference between var, let, and const**.

· Explain **ES6 class fundamentals and inheritance**.

· Use **map()** to display player details.

· Apply **arrow functions** for filtering.

· Implement **array destructuring and merging** using ES6.

· Display multiple components conditionally using a **flag variable**.

**Implementation:**

**Step 1: Create a React App**

****

**Step 2: Create Component and CSS File**

Inside the src/Components/ folder:

**ListofPlayers.js**

· Declare an array of **11 players with name and score**.

· Use map() to render the list.

· Filter players with score below 70 using **arrow functions**.

import React from 'react';

function ListofPlayers() {

const players = [

{ name: "Virat Kohli", score: 85 },

{ name: "Rohit Sharma", score: 95 },

{ name: "KL Rahul", score: 65 },

{ name: "Hardik Pandya", score: 72 },

{ name: "Rishabh Pant", score: 50 },

{ name: "Shreyas Iyer", score: 40 },

{ name: "Ravindra Jadeja", score: 35 },

{ name: "Bumrah", score: 15 },

{ name: "Shami", score: 20 },

{ name: "Kuldeep", score: 30 },

{ name: "Ashwin", score: 25 }

];

const filteredPlayers = players.filter(p => p.score < 70); // Arrow Function

return (

<div>

<h2>All Players</h2>

<ul>

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

<li key={index}>{player.name} - {player.score}</li>

))}

</ul>

<h3>Players with score below 70</h3>

<ul>

{filteredPlayers.map((player, index) => (

<li key={index}>{player.name} - {player.score}</li>

))}

</ul>

</div>

);

}

export default ListofPlayers;

**IndianPlayers.jsx**

import React from 'react';

function IndianPlayers() {

const oddTeam = ["Virat", "Rohit", "Rahul"];

const evenTeam = ["Jadeja", "Bumrah", "Shami"];

const [player1, player2, player3] = oddTeam;

const T20players = ["Dhoni", "Kohli", "Raina"];

const RanjiPlayers = ["Pujara", "Rahane", "Karun"];

const allPlayers = [...T20players, ...RanjiPlayers];

return (

<div>

<h2>Odd Team Players: {player1}, {player2}, {player3}</h2>

<h2>Even Team Players: {evenTeam.join(", ")}</h2>

<h3>Merged Players List:</h3>

<p>{allPlayers.join(", ")}</p>

</div>

);

}

export default IndianPlayers;

**Step 3: Use the Component in App.js**

import React from 'react';

import ListofPlayers from './Components/ListofPlayers';

import IndianPlayers from './Components/IndianPlayers';

function App() {

const flag = true; // Change to false to see other component

return (

<div className="App">

<h1>Cricket Application</h1>

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

</div>

);

}

export default App;

**Step 4: Run the App**

****

**Output:**

When const flag=true, in App.js:

When const flag=false, in App.js:

