## **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.

**Output:**

When Flag=true



When Flag=false



**Hint:**



**Program:**

**IndianPlayers.js:**

import React from "react";

const IndianPlayers = () => {

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

const RanjiTrophyPlayers = ["Pujara", "Rahane", "Nair", "Saha"];

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

const oddTeam = allPlayers.filter((\_, index) => index % 2 !== 0);

const evenTeam = allPlayers.filter((\_, index) => index % 2 === 0);

return (

<div>

<h2>Odd Team Players</h2>

<ul>

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

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

))}

</ul>

<h2>Even Team Players</h2>

<ul>

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

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

))}

</ul>

</div>

);

};

export default IndianPlayers;

**ListofPlayers.js:**

import React from "react";

const ListofPlayers = () => {

const players = [

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

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

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

{ name: "Pant", score: 45 },

{ name: "Shami", score: 75 },

{ name: "Bumrah", score: 60 },

{ name: "Ashwin", score: 88 },

{ name: "Jadeja", score: 55 },

{ name: "Kohli", score: 99 },

{ name: "Gill", score: 66 },

{ name: "Surya", score: 70 },

];

const highScorers = players.filter((player) => player.score >= 70);

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

return (

<div>

<h2>High Scoring Players (Score ≥ 70)</h2>

<ul>

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

<li key={index}>

{player.name} - {player.score}

</li>

))}

</ul>

<h2>Low Scoring Players (Score &lt; 70)</h2>

<ul>

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

<li key={index}>

{player.name} - {player.score}

</li>

))}

</ul>

</div>

);

};

export default ListofPlayers;

**App.js:**

import React from 'react';

import ListofPlayers from './components/ListofPlayers';

import IndianPlayers from './components/IndianPlayers';

function App() {

const flag = true;

return (

<div className="App">

<h1>🏏 Cricket App</h1>

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

</div>

);

}

export default App;

**index.js:**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(<App />);

**package.json:**

{

"name": "cricketapp",

"version": "0.1.0",

"private": true,

"dependencies": {

"@testing-library/jest-dom": "^5.17.0",

"@testing-library/react": "^14.0.0",

"@testing-library/user-event": "^14.4.3",

"react": "^18.2.0",

"react-dom": "^18.2.0",

"react-scripts": "5.0.1",

"web-vitals": "^2.1.4"

},

"scripts": {

"start": "react-scripts start",

"build": "react-scripts build",

"test": "react-scripts test",

"eject": "react-scripts eject"

},

"eslintConfig": {

"extends": [

"react-app",

"react-app/jest"

]

},

"browserslist": {

"production": [

">0.2%",

"not dead",

"not op\_mini all"

],

"development": [

"last 1 chrome version",

"last 1 firefox version",

"last 1 safari version"

]

}

}

**pom.xml:**

<project>

<modelVersion>4.0.0</modelVersion>

<groupId>mygroupid</groupId>

<artifactId>myartifactid</artifactId>

<version>0.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<type>jar</type>

</dependency>

</dependencies>

</project>

**.replit:**

run = "npm start"

modules = ["nodejs-20"]

[[ports]]

localPort = 3000

externalPort = 80

**Output:**







