**WEEK-7 ReactJS-HOL**

**SUPERSETID 6393674**

## **Objectives**

**• List the features of ES6**

ECMAScript 6 (ES6) introduced powerful features such as let and const, arrow functions, template literals, default parameters, spread/rest operators, destructuring, promises, classes, modules, and enhanced object literals. These features improve code readability, maintainability, and performance.

**• Explain JavaScript let**

The let keyword declares block-scoped variables, meaning they are only accessible within the block or function where they are defined. Unlike var, let does not allow re-declaration in the same scope and helps prevent common bugs due to scoping issues.

**• Identify the differences between var and let**

\*var is function-scoped, while let is block-scoped.

\* Variables declared with var are hoisted and can be accessed before declaration (with undefined), while let throws an error if accessed before declaration.

\*let prevents re-declaration in the same scope; var does not

**• Explain JavaScript const**

The const keyword is used to declare variables with a constant value. It also has block scope like let. A const variable must be initialized during declaration, and its reference cannot be changed (though the contents of arrays or objects declared with const can still be modified).

**• Explain ES6 class fundamentals**

ES6 introduced a more intuitive syntax for creating classes. Classes in ES6 are syntactic sugar over JavaScript's existing prototype-based inheritance. They support constructors, methods, and can be instantiated using the new keyword.

**• Explain ES6 class inheritance**

ES6 classes support inheritance using the extends keyword. A child class can inherit properties and methods from a parent class. The super() function is used to call the constructor of the parent class within the child class.

**• Define ES6 arrow functions**

Arrow functions provide a shorter syntax for writing function expressions. They do not bind their own this, instead they inherit this from the surrounding context.

**SYNTAX:**

**const sum = (a, b) => a + b;**

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.

**APP.js**

import React from 'react';

import EvenPlayers from './components/EvenPlayers';

import ListofIndianPlayers from './components/ListofIndianPlayers';

import ListofPlayers from './components/ListofPlayers';

import OddPlayers from './components/OddPlayers';

import ScoreBelow70 from './components/ScoreBelow70';

function App() {

  var flag = true;

  const players = [

    { name: "Jack", score: 50 },

    { name: "Michael", score: 70 },

    { name: "John", score: 40 },

    { name: "Ann", score: 61 },

    { name: "Elisabeth", score: 61 },

    { name: "Sachin", score: 95 },

    { name: "Dhoni", score: 100 },

    { name: "Virat", score: 84 },

    { name: "Jadeja", score: 64 },

    { name: "Raina", score: 75 },

    { name: "Rohit", score: 80 }

  ];

  const IndianTeam = ["Sachin1", "Dhoni2", "Virat3", "Rohit4", "Yuvaraj5", "Raina6"];

  const IndianPlayers = ["First Player", "Second Player", "Third Player", "Fourth Player", "Fifth Player", "Sixth Player"];

  if (flag === true) {

    return (

      <div>

        <h1>List of Players</h1>

        <ListofPlayers players={players} />

        <hr />

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

        <ScoreBelow70 players={players} />

      </div>

    );

  } else {

    return (

      <div>

        <div>

          <h1>Indian Team</h1>

          <h1>Odd Players</h1>

          <OddPlayers team={IndianTeam} />

          <hr />

          <h1>Even Players</h1>

          <EvenPlayers team={IndianTeam} />

        </div>

        <hr />

        <div>

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

          <ListofIndianPlayers IndianPlayers={IndianPlayers} />

        </div>

      </div>

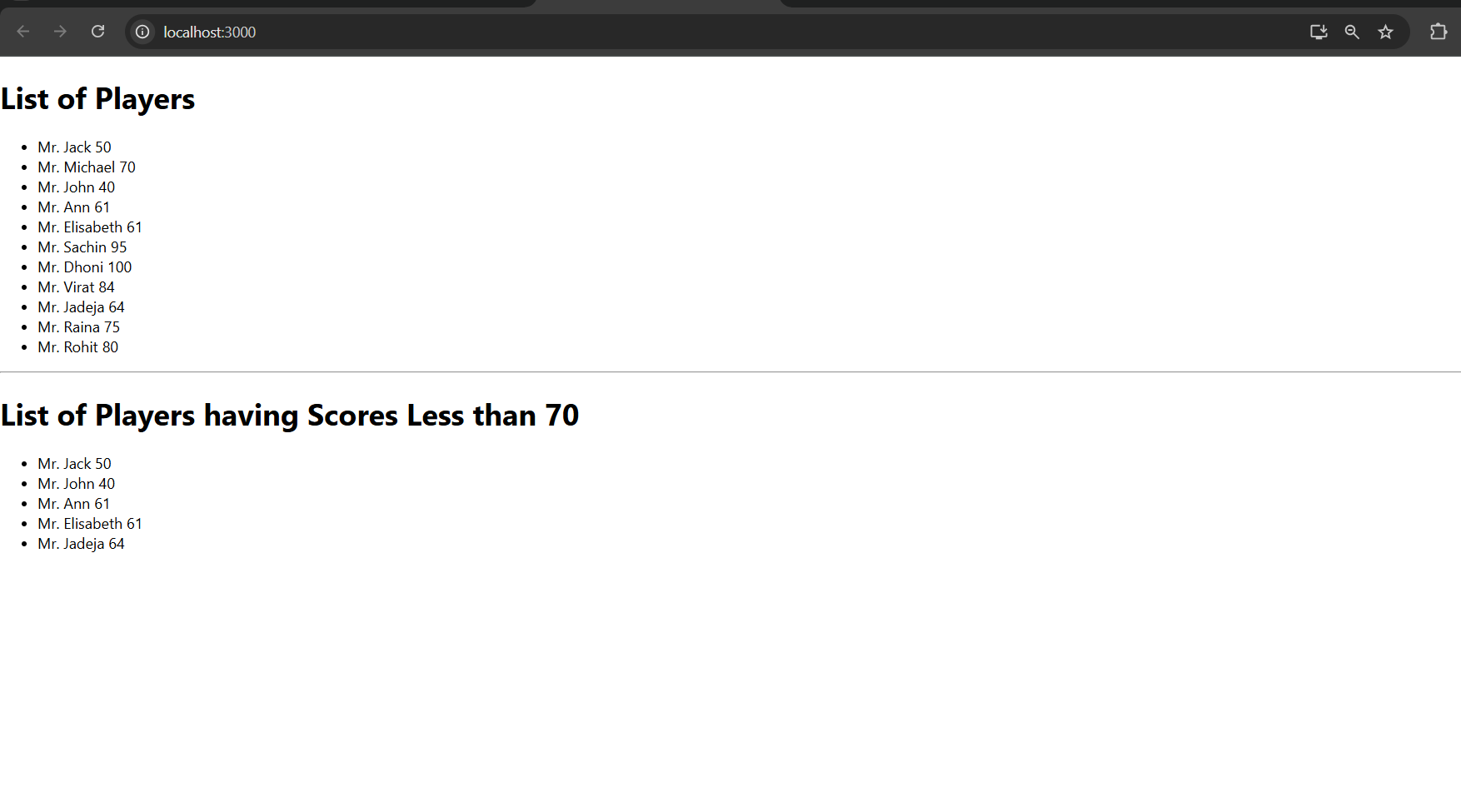
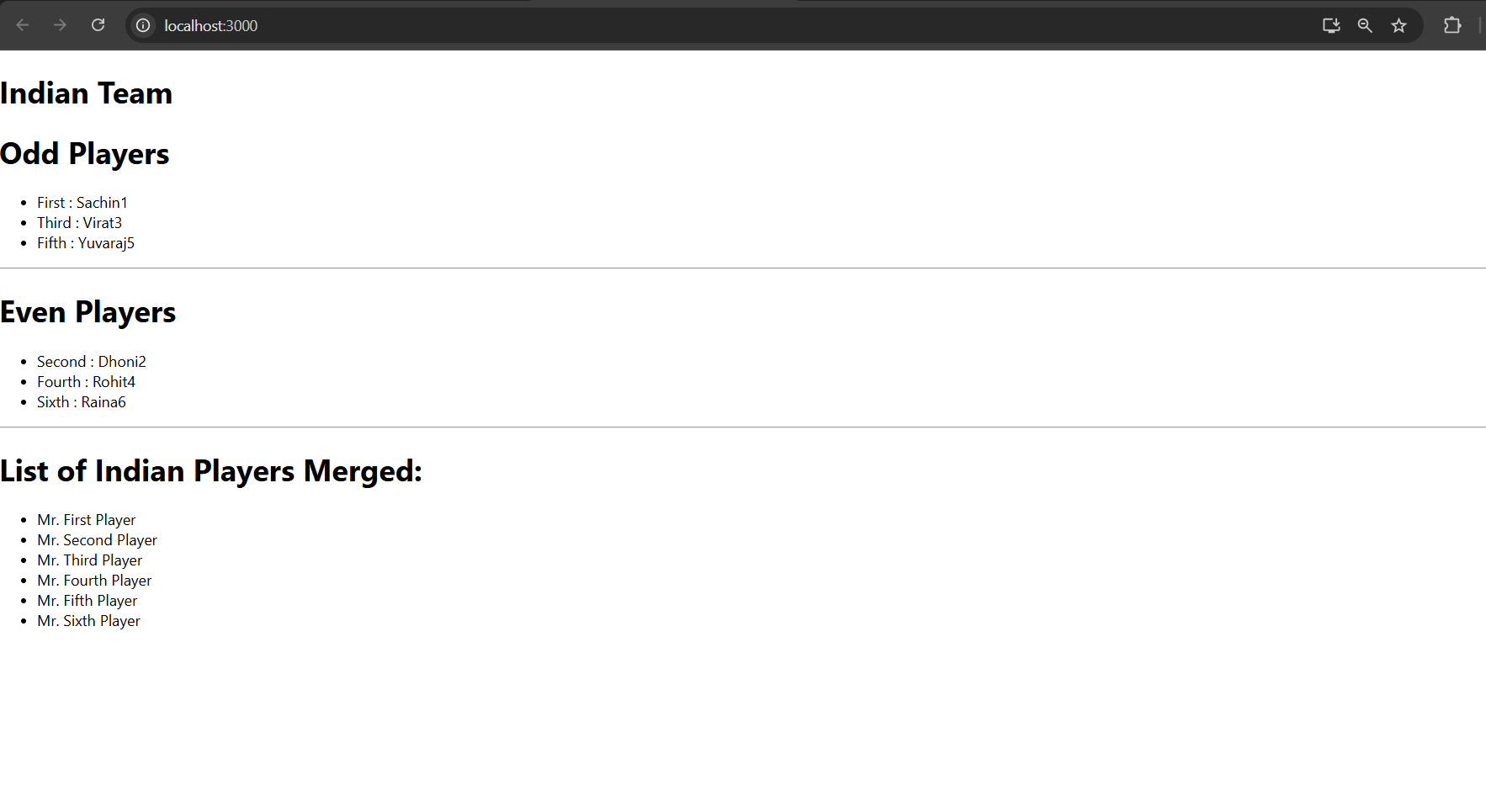
    );

  }

}

export default App;

**OUTPUT**

****