**9. ReactJS-HOL**

**Objective:**

Q1. List the features of ES6

* let and const declarations
* Arrow functions
* Classes and Inheritance
* Default parameters
* Template literals
* Destructuring assignment
* Spread and Rest operators
* Promises
* Modules (import/export)
* Map and Set objects

Q2. Explain JavaScript let

let is a block-scoped variable declaration introduced in ES6. It allows you to declare variables limited to the scope of a block, statement, or expression, unlike var which is function-scoped.

Q3. Identify the differences between var and let

| **Feature** | var | let |
| --- | --- | --- |
| **Scope** | Function-scoped | Block-scoped |
| **Hoisting** | Hoisted (initialized as undefined) | Hoisted (but not initialized) |
| **Redeclaration** | Allowed | Not allowed in same scope |
| **Global object** | Becomes a property | Does not become a property |

Q4. Explain JavaScript const

const is used to declare constants. A variable declared with const must be assigned at the time of declaration and cannot be reassigned. However, for objects and arrays, the contents (properties or elements) can still be modified.

Q5. Explain ES6 class fundamentals

ES6 introduced the class syntax to define constructor functions and prototypes more cleanly. A class can have a constructor, methods, and can be instantiated using the new keyword.

Example:

class Person {

constructor(name) {

this.name = name;

}

greet() {

return `Hello, ${this.name}`;

}

}

Q6. Explain ES6 class inheritance

Inheritance in ES6 is achieved using the extends keyword. A subclass can inherit properties and methods from a parent class.

Example:

class Animal {

speak() {

return "Animal speaks";

}

}

class Dog extends Animal {

bark() {

return "Dog barks";

}

}

Q7. Define ES6 arrow functions

Arrow functions provide a concise syntax for writing function expressions and do not bind their own this.

Syntax:

const add = (a, b) => a + b;

Q8. Identify set(), map()

Set: A collection of unique values. No duplicates allowed.

const numbers = new Set([1, 2, 3]);

Map: A key-value collection where keys can be of any data type.

const scores = new Map();

scores.set('John', 90);

**Hands-On exercise:**

**Code:**

**App.js**

import React from 'react';  
import './App.css';  
import ListofPlayers from './ListofPlayers';  
import Scorebelow70 from './Scorebelow70';  
import { OddPlayers } from './OddPlayers';  
import { EvenPlayers } from './EvenPlayers';  
import ListofIndianPlayers from './ListofIndianPlayers';  
  
function App() {  
 var flag = false; // Change to true or false to toggle output  
  
 const players = [  
 { name: 'Jack', score: 50 },  
 { name: 'Michael', score: 70 },  
 { name: 'John', score: 40 },  
 { name: 'Aiden', score: 61 },  
 { name: 'Elizabeth', score: 61 },  
 { name: 'Sachin', score: 95 },  
 { name: 'Virat', score: 90 },  
 { name: 'Jadeja', score: 64 },  
 { name: 'Rain', score: 75 },  
 { name: 'Rohit', score: 70 },  
 { name: 'Dhoni', score: 85 }  
 ];  
  
 const T20Players = ['First Player', 'Second Player', 'Third Player'];  
 const RanjiTrophyPlayers = ['Fourth Player', 'Fifth Player', 'Sixth Player'];  
 const IndianPlayers = [...T20Players, ...RanjiTrophyPlayers];  
  
 const IndianTeam = ['Sachin', 'Dhoni', 'Virat', 'Rohit', 'Yuvraj', 'Rahul'];  
  
 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(IndianTeam)}  
 <hr />  
 <h1> Even Players </h1>  
 {EvenPlayers(IndianTeam)}  
 </div>  
 <hr />  
 <div>  
 <h1> List of Indian Players Merged:</h1>  
 <ListofIndianPlayers IndianPlayers={IndianPlayers} />  
 </div>  
 </div>  
 );  
 }  
}  
  
export default App;

**ListofPlayers.js**

import React from 'react';  
  
function ListofPlayers({ players }) {  
 return (  
 <ul>  
 {players.map((item) => (  
 <li key={item.name}>  
 Mr. {item.name} <span> {item.score} </span>  
 </li>  
 ))}  
 </ul>  
 );  
}  
  
export default ListofPlayers;

**Scorebelow70.js**

import React from 'react';  
  
function Scorebelow70({ players }) {  
 let players70 = [];  
 players.map((item) => {  
 if (item.score <= 70) {  
 players70.push(item);  
 }  
 });  
  
 return (  
 <ul>  
 {players70.map((item) => (  
 <li key={item.name}>  
 Mr. {item.name} <span> {item.score} </span>  
 </li>  
 ))}  
 </ul>  
 );  
}  
  
export default Scorebelow70;

**OddPlayers.js**

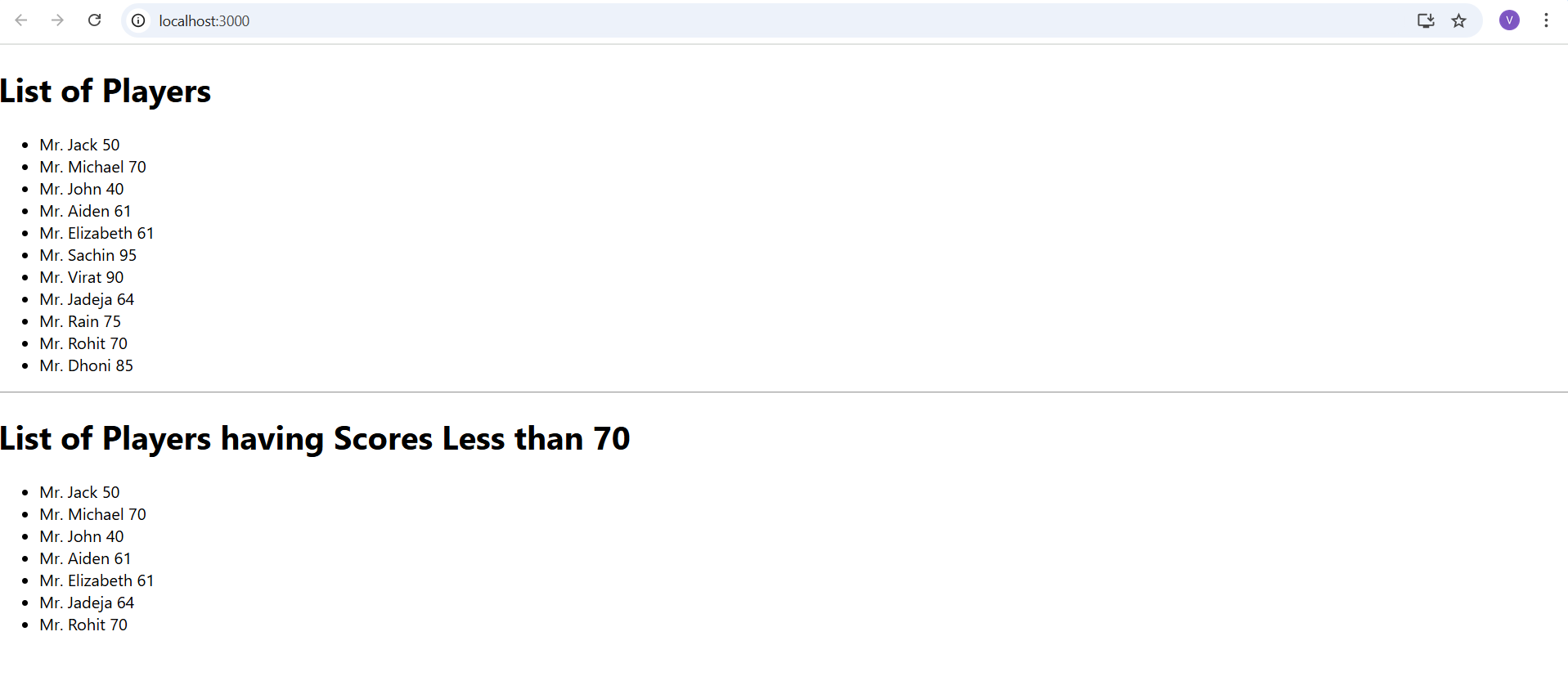
import React from 'react';  
  
export function OddPlayers([first, , third, , fifth]) {  
 return (  
 <div>  
 <li> First : {first} </li>  
 <li> Third : {third} </li>  
 <li> Fifth : {fifth} </li>  
 </div>  
 );  
}

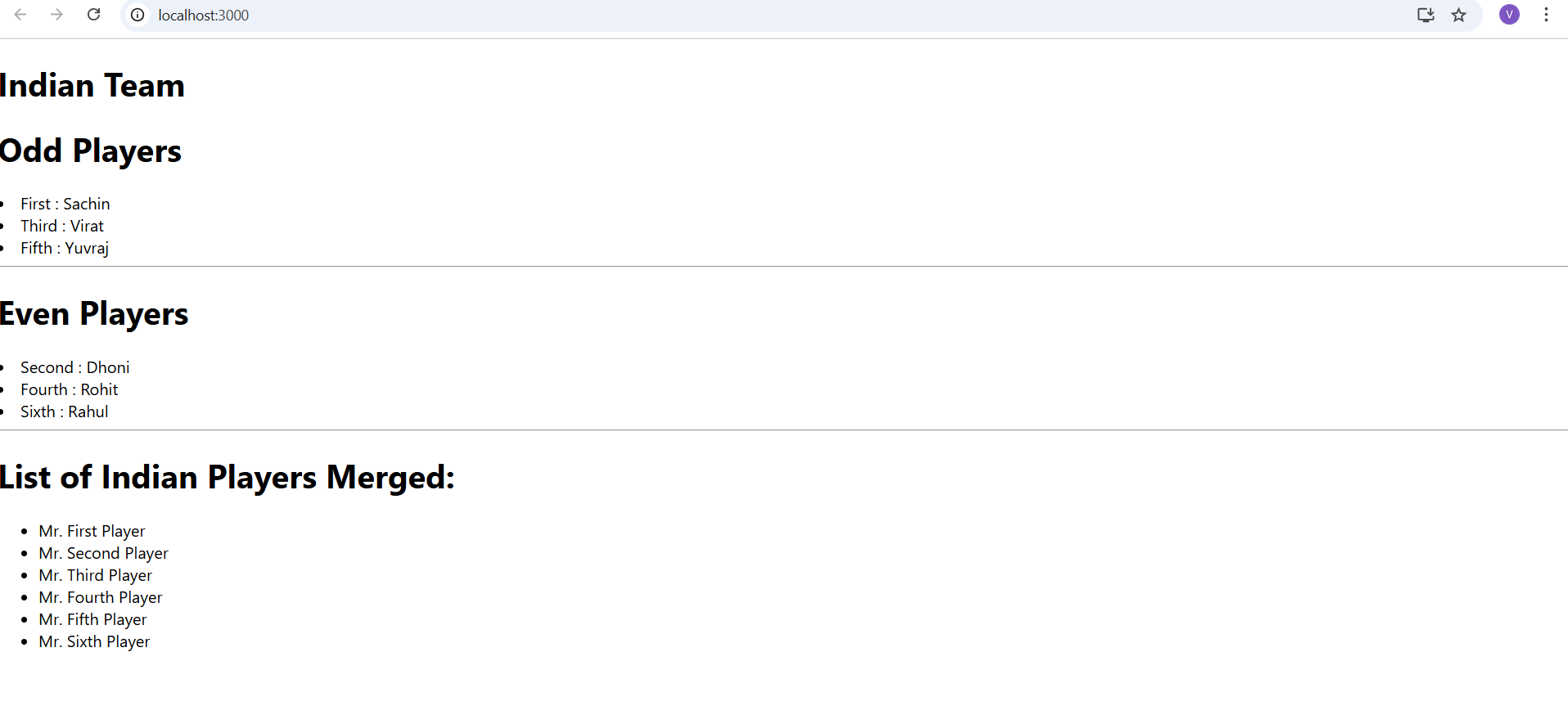
**EvenPlayers.js**

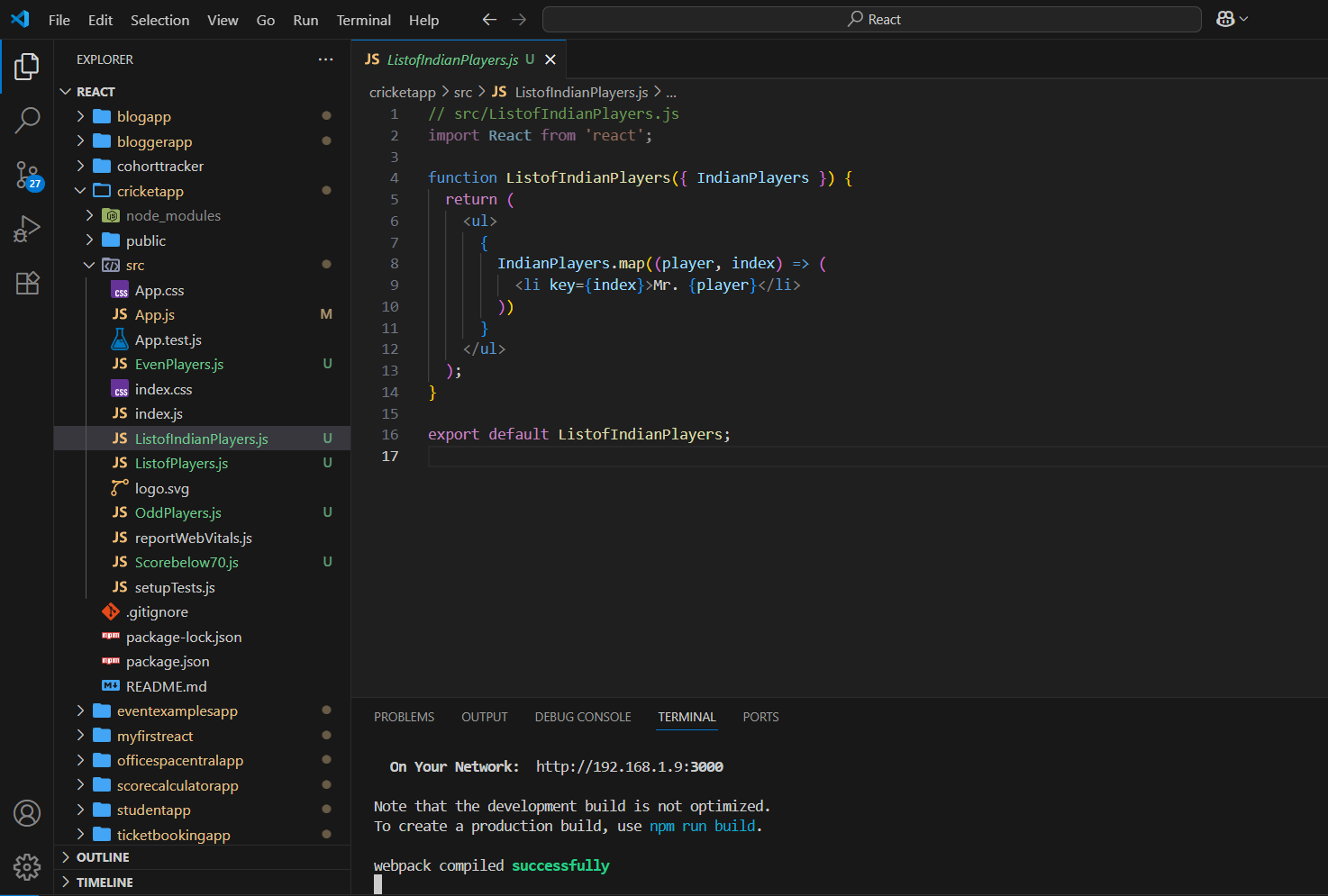
import React from 'react';  
  
export function EvenPlayers([ , second, , fourth, , sixth]) {  
 return (  
 <div>  
 <li> Second : {second} </li>  
 <li> Fourth : {fourth} </li>  
 <li> Sixth : {sixth} </li>  
 </div>  
 );  
}

**ListofIndianPlayers.js**

import React from 'react';  
  
function ListofIndianPlayers({ IndianPlayers }) {  
 return (  
 <ul>  
 {IndianPlayers.map((player, index) => (  
 <li key={index}>Mr. {player}</li>  
 ))}  
 </ul>  
 );  
}  
  
export default ListofIndianPlayers;

**Output:  
**

****

****