**9. ReactJS – HOL**

**Objectives**

* **List the features of ES6**

1. let and const for block-scoped variables (replacing var)
2. **Arrow functions** (()=>{}) – shorter syntax for writing functions
3. **Template literals** using backticks (`Hello, ${name}`)
4. **Destructuring** of arrays and objects
5. **Default parameters** in functions
6. **Spread operator (**...**)** to expand arrays or objects
7. **Rest parameters (**...**)** to group multiple arguments into one array

* **Explain JavaScript let**

let is used to declare variables in JavaScript. It was introduced in ES6 (ECMAScript 2015). It allows you to create variables that can be updated, but not re-declared in the same scope.  
  
For example:  
let name = "Vika";  
console.log(name);   
name = "Sita"; //reassigning  
console.log(name);   
let name = "John"; // error

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

|  |  |  |
| --- | --- | --- |
| **Feature** | **var** | **let** |
| Scope | Function-scoped | Block-scoped ({}) |
| Redeclaration | Allowed within the same scope | Not allowed in the same scope |
| Hoisting | Hoisted and initialized as undefined | Hoisted but not initialized |
| Usage Before Declaration | Allowed (but gets undefined) | Not allowed (throws error) |
| Global Object Property | Yes (if declared globally, becomes a property of window) | No |

* **Explain JavaScript const**

const is used to **declare variables** whose **value cannot be changed** after it's assigned.It was introduced in **ES6 (ECMAScript 2015).**It is **block-scoped**, just like let

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Constant value | Once assigned, the value cannot be changed |
| Block-scoped | Only available inside the {} where it's defined |
| Must be initialized | You **must assign a value** at the time of declaration |

* **Explain ES6 class fundamentals**  
  A class is a blueprint for creating objects with shared properties and methods

Features:

1. class - Keyword to define a class
2. constructor() - Special method to set initial values
3. this - Refers to the current object instance
4. Methods - Functions inside the class
5. new keyword - Used to create an instance (object) of the class

* **Explain ES6 class inheritance**

In ES6, inheritance allows one class to use properties and methods of another class.  
It helps you avoid repeating code by creating a child class (subclass) from a parent class (superclass).

Keyword:

1. extends - Used to inherit from another class
2. super() - Used inside the constructor to call the parent class constructor

* **Define ES6 arrow functions**

Arrow functions are a shorter way to write functions in JavaScript.

Example:

*// Traditional function*

*function add(a, b) {  
 return a + b;  
}*

*// Arrow function*

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

* **Identify set(), map()**

1. A Set is a collection of unique values — no duplicates allowed.

*Syntax: const mySet = new Set();*

Features of Set:

* *Stores* ***unique*** *values only.*
* *Can store any data type: numbers, strings, objects.*
* *Methods: .add(), .delete(), .has(), .clear()*

2. A Map is a collection of key-value pairs, where keys can be of any data type

*Syntax: const myMap = new Map();*

*Features of Map:*

* *Keys can be* ***any type*** *(object, number, etc.)*
* *Keeps the* ***order of insertion***
* *Methods: .set(), .get(), .has(), .delete(), .clear()*

**EXERCISE – 2:** Creation of a Cricket App

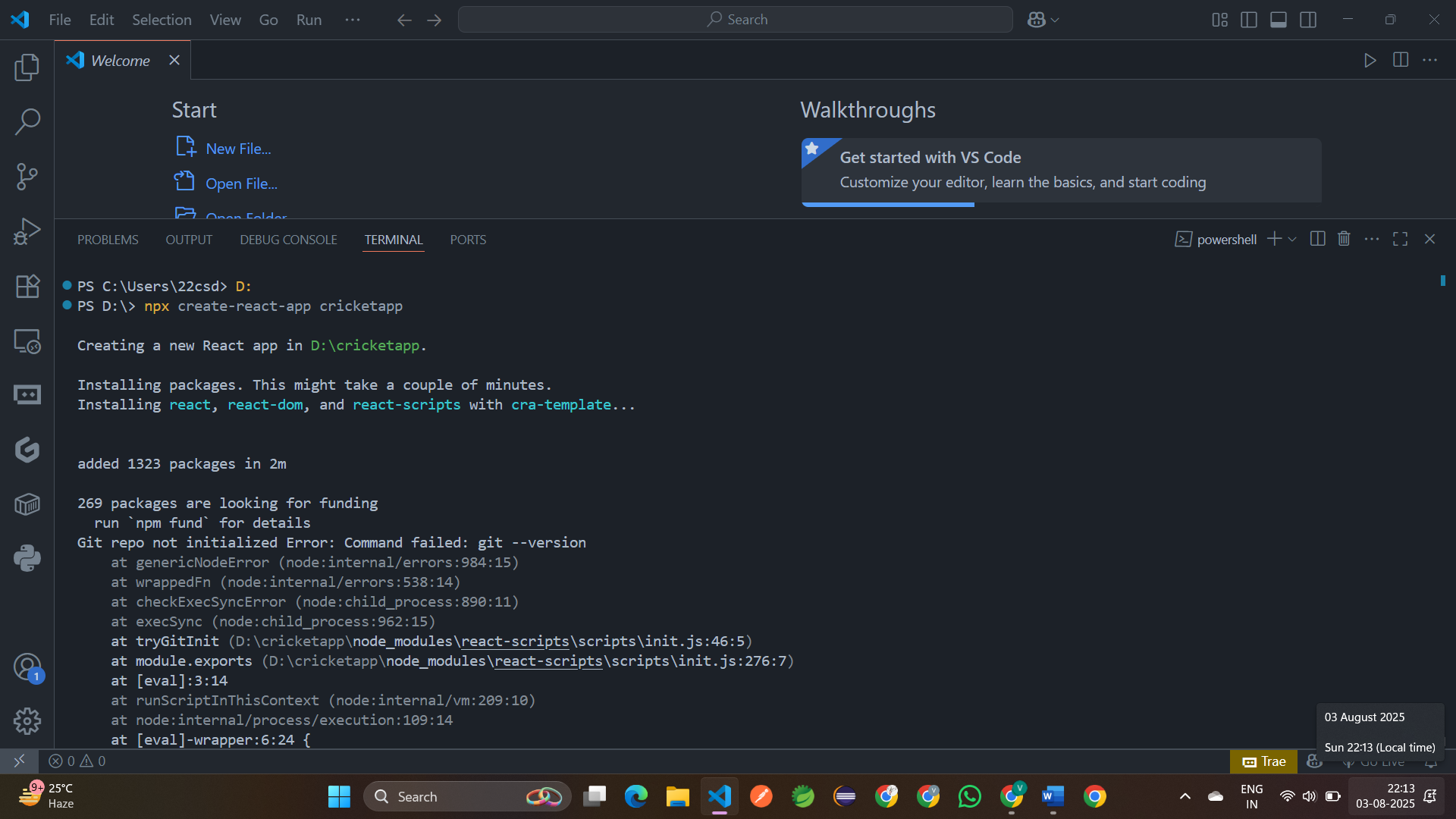
Step 1: Open **VS Code**

Step 2: Create a New React App

**npx create-react-app cricketapp**

Step 3: Navigate into the project folder

**cd cricketapp**



Step 4: Create a new file **ListofPlayers.js**

import React from 'react';  
 const players = [ { name: 'Virat', score: 100 }, { name: 'Rohit', score: 60 },  
 { name: 'Rahul', score: 55 }, { name: 'Hardik', score: 74 },  
 { name: 'Rishabh', score: 88 }, { name: 'Shami', score: 40 },  
 { name: 'Bumrah', score: 65 }, { name: 'Ashwin', score: 50 },  
 { name: 'Surya', score: 91 }, { name: 'Ishan', score: 59 },  
 { name: 'Gill', score: 45 }];

export default function ListofPlayers() { // 1. Display all players  
 const allPlayers = players.map((item, idx) => (  
 <li key={idx}>   
 Mr. {item.name}: <span>{item.score}</span> </li> ));  
 // 2. Display players with score <= 70  
 const filteredPlayers = players  
 .filter(item => item.score <= 70)  
 .map((item, idx) => (  
 <li key={idx}>  
 Mr. {item.name}: <span>{item.score}</span>  
 </li> ));  
 return (  
 <div>  
 <h2>All Players</h2>  
 <ul>{allPlayers}</ul>  
 <h2>List of Players having Scores less than 70</h2>  
 <ul>{filteredPlayers}</ul>  
 </div> );  
 }

Step 6: Create a new file **IndianPlayers.js**

import React from 'react';  
 const T20Players = ['Sachin1', 'Dhoni2', 'Virat3'];  
 const RanjiTrophyPlayers = ['Rohit4', 'Yuvaraj5', 'Raina6'];  
 const IndianPlayers = [...T20Players, ...RanjiTrophyPlayers];  
 export function OddPlayers([first, , third, , fifth]) {  
   return (  
     <div>      <ul>  
         <li>First: {first}</li>  
         <li>Third: {third}</li>  
         <li>Fifth: {fifth}</li>  
       </ul>  
     </div>  
   );}  
 export function EvenPlayers([, second, , fourth, , sixth]) {

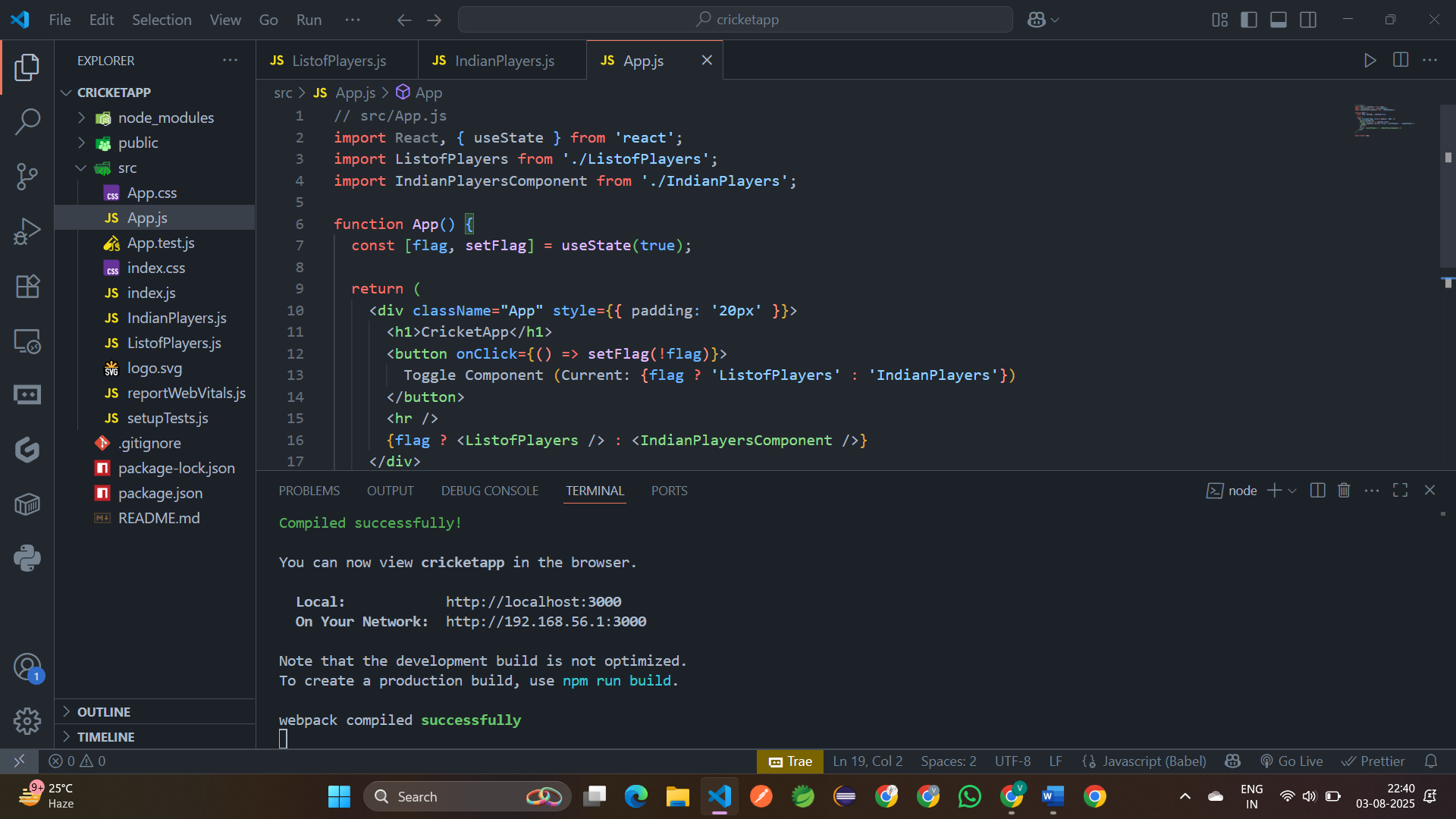
  return (  
     <div>      <ul>  
         <li>Second: {second}</li>  
         <li>Fourth: {fourth}</li>  
         <li>Sixth: {sixth}</li>  
       </ul>    </div>  
   );}  
 export default function IndianPlayersComponent() {  
   return (  
     <div>  
       <h2>Merged Indian Players</h2>  
       <ul>        {IndianPlayers.map((player, idx) => (  
           <li key={idx}>{player}</li>        ))}  
       </ul>  
       <h3>Odd Team Players</h3>  
       <OddPlayers {...[IndianPlayers]} />  
       <h3>Even Team Players</h3>  
       <EvenPlayers {...[IndianPlayers]} />  
     </div>  );}

step 7: Open **src/App.js** and modify it

import React, { useState } from 'react';  
 import ListofPlayers from './ListofPlayers';  
 import IndianPlayersComponent from './IndianPlayers';  
 function App() {  
 const [flag, setFlag] = useState(true);  
 return (  
 <div className="App" style={{ padding: '20px' }}>   
 <h1>CricketApp</h1>   
 <button onClick={() => setFlag(!flag)}> Toggle Component (Current: {flag ?   
 'ListofPlayers' : 'IndianPlayers'})  
 </button> <hr /> {flag ? <ListofPlayers /> : <IndianPlayersComponent />} </div> ); }  
 export default App;

step 8: Run the App using **npm start**

Url: <http://localhost:3000>



**Output:**

