

Digital Nurture 7.0 Deep Skilling

Week 7

Hands on 1 : Create a React Application named “cricket app” with the following components:

1. List of Players

- 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.

2. Indian Players

- a. Display the Odd Team Player and Even Team players using the Destructuring features of ES6
- b. Declare two arrays T20players and Ranji Trophy 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.

```
import React, { useState } from 'react';
import ListofPlayers from './Components/ListofPlayers';
import IndianPlayers from './Components/IndianPlayers';
import './App.css';
import './Components/ComponentStyles.css';

function App() {
  const [flag, setFlag] = useState(true);

  return (
    <div className="App">
      <h1> 🏏 Cricket App </h1>
      <button className="toggle-btn" onClick={() => setFlag(!flag)}>
        {flag ? 'View Indian Players' : 'View Player List'}
      </button>

      <div className="component-container">
        {flag ? <ListofPlayers /> : <IndianPlayers />}
      </div>
    </div>
  );
}

export default App;
```

```

const lowScorers = players.filter(player => player.score < 70);
return (
  <>
    <h2>All Players</h2>
    <table className="player-table">
      <thead>
        <tr>
          <th>Player Name</th>
          <th>Score</th>
        </tr>
      </thead>
      <tbody>
        {players.map((player, index) => (
          <tr key={index}>
            <td>{player.name}</td>
            <td>{player.score}</td>
          </tr>
        ))}
      </tbody>
    </table>
  </>
)

```

```

const IndianPlayers = () => {
  const team = ['Virat', 'Rohit', 'Rahul', 'Gill', 'Hardik', 'Jadeja', 'Bumrah', 'Shami'];
  const oddPlayers = team.filter((_, index) => index % 2 === 0);
  const evenPlayers = team.filter((_, index) => index % 2 !== 0);
  const T20Players = ['Rohit', 'Gill', 'Surya'];
  const RanjiTrophyPlayers = ['Pujara', 'Rahane', 'Karun'];
  const allPlayers = [...T20Players, ...RanjiTrophyPlayers];
  return (
    <>
      <h2>Odd Team Players</h2>
      <ul className="player-list">
        {oddPlayers.map((player, index) => <li key={index}>{player}</li>)}
      </ul>

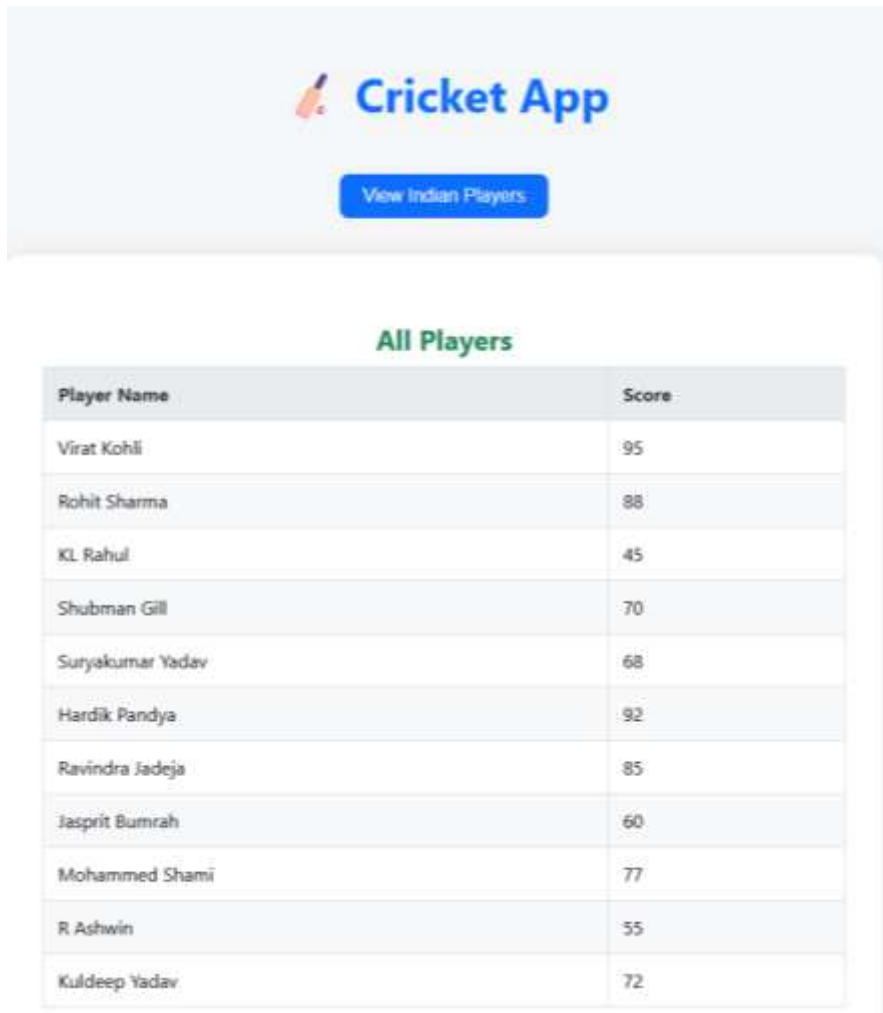
      <h2>Even Team Players</h2>
      <ul className="player-list">
        {evenPlayers.map((player, index) => <li key={index}>{player}</li>)}
      </ul>

      <h2>Merged T20 + Ranji Trophy Players</h2>
      <ul className="player-list">
        {allPlayers.map((player, index) => <li key={index}>{player}</li>)}
      </ul>
    </>
  );
};

```

Output :

When flag = true



The image shows a web application interface for a 'Cricket App'. At the top, there is a header with a cricket bat icon and the text 'Cricket App'. Below the header is a blue button labeled 'View Indian Players'. The main content area is titled 'All Players' in green. It contains a table with two columns: 'Player Name' and 'Score'. The table lists 11 players with their respective scores.

Player Name	Score
Virat Kohli	95
Rohit Sharma	88
KL Rahul	45
Shubman Gill	70
Suryakumar Yadav	68
Hardik Pandya	92
Ravindra Jadeja	85
Jasprit Bumrah	60
Mohammed Shami	77
R Ashwin	55
Kuldeep Yadav	72

Players with Score Below 70

Player Name	Score
KL Rahul	45
Suryakumar Yadav	68
Jasprit Bumrah	60
R Ashwin	55

When flag=false



Cricket App

[View Player List](#)

Odd Team Players

Virat

Rahul

Hardik

Bumrah

Even Team Players

Rohit

Gill

Jadeja

Shami

Merged T20 + Ranji Trophy Players

Rohit

Gill

Surya

Pujara

Rahane

Karun