WEEK:7

1. List the features of ES6

 let and const for block-scoped variables.

 Arrow functions (=>)

 Classes (class, constructor)

 Default parameters in functions

1. Explain JavaScript let

 let is used to declare **block-scoped** variables.

 Unlike var, let does not allow re-declaration within the same scope.

1. Identify the differences between var and let

var is function-scoped, so it's accessible throughout the function in which it's declared.

let is block-scoped, meaning it's only accessible within the {} block where it's defined.

1. Explain JavaScript const

 const declares a **block-scoped constant**.

 Once assigned, the value **cannot be reassigned**.

1. Explain ES6 class fundamentals

Includes a constructor() method and other methods defined without the function keyword

1. Explain ES6 class inheritance

 Use the extends keyword for inheritance.

 Use super() to call the parent class's constructor.

1. Define ES6 arrow functions

 oncise syntax: () => {}.

 No binding of this, arguments, or super.

 Cannot be used as constructors.

1. ES6 introduces a clean syntax to define classes using the class keyword.

Identify set(), map()

**Set:**

* A collection of **unique values**.
* Duplicate entries are ignored.

**Example:**

javascript

CopyEdit

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

console.log(numbers);

**Map:**

* A collection of key-value pairs.
* Keys can be **any type** (objects, functions, etc.).

**Example:**

javascript

CopyEdit

const map = new Map();

map.set("name", "Alice");

Create a React Application named “cricketapp” with the following components:

ListofPlayers.js:

import React from 'react';

const ListofPlayers = () => {

  const players = [

    { name: "Virat", score: 95 },

    { name: "Rohit", score: 88 },

    { name: "Rahul", score: 45 },

    { name: "Pant", score: 73 },

    { name: "Hardik", score: 55 },

    { name: "Jadeja", score: 92 },

    { name: "Ashwin", score: 67 },

    { name: "Bumrah", score: 81 },

    { name: "Shami", score: 61 },

    { name: "Dhoni", score: 90 },

    { name: "Gill", score: 70 }

  ];

  const allPlayers = players.map((player, index) => (

    <li key={index}>{player.name} - {player.score}</li>

  ));

  const below70 = players.filter(p => p.score < 70).map((p, index) => (

    <li key={index}>{p.name} - {p.score}</li>

  ));

  return (

    <div>

      <h2>All Players:</h2>

      <ul>{allPlayers}</ul>

      <h2>Players with score below 70:</h2>

      <ul>{below70}</ul>

    </div>

  );

};

export default ListofPlayers;

IndianPlayers.js:

import React from 'react';

const IndianPlayers = () => {

  const players = ["Virat", "Rohit", "Rahul", "Pant", "Hardik", "Jadeja"];

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

  const evenTeam = players.filter((\_, i) => i % 2 === 0);

  const T20Players = ["Surya", "Ishan"];

  const RanjiTrophyPlayers = ["Shreyas", "Sarfaraz"];

  const mergedPlayers = [...T20Players, ...RanjiTrophyPlayers];

  return (

    <div>

      <h2>Odd Team Players:</h2>

      <ul>{oddTeam.map((p, i) => <li key={i}>{p}</li>)}</ul>

      <h2>Even Team Players:</h2>

      <ul>{evenTeam.map((p, i) => <li key={i}>{p}</li>)}</ul>

      <h2>Merged T20 & Ranji Trophy Players:</h2>

      <ul>{mergedPlayers.map((p, i) => <li key={i}>{p}</li>)}</ul>

    </div>

  );

};

export default IndianPlayers;

App.js:

import React from 'react';

import ListofPlayers from './components/ListofPlayers';

import IndianPlayers from './components/IndianPlayers';

function App() {

  const flag = true; // Toggle this to false to change view

  return (

    <div className="App">

      <h1>🏏 Cricket App</h1>

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

    </div>

  );

}

export default App;

Output:

