## **Objectives**

**1.List the features of ES6**

ES6, also known as ECMAScript 2015, introduced many new features that made JavaScript more powerful and easier to write. Some the key feature include:

* Introduction of “let” and “const” for better variable scoping.
* Arrow functions for shorter and cleaner function syntax.
* Classes for writing object-oriented code in a simpler way.
* Template literals for easier string formatting.
* Default parameters in functions.
* Spread and rest operators for handling arrays and function arguments.
* Modules to import and export code across files.
* Promises for handling asynchronous operations more clearly.
* New data structures like “Set” and “Map”.

**2.Explain JavaScript let**

The “let” keyword is used to declare variables in JavaScript. Unlike older declaration, variables declared with “let” are block-scoped, meaning they only exist within the block they are defined in. This helps avoid problems that occur due to variable redeclaration or accidental overwrites. It is commonly used in modern JavaScript code.

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

The key differences between “var” and “let” are:

* “var” is function-scoped, while “let” is block-scoped.
* Variables declared with “var” are hoisted and initialized with “undefined”, whereas “let” variables are hoisted but not initialized.
* “let” does not allow re-declaring the same variable in the same scope, but “var” does.
* Using “let” provides better control over variable usage and reduces bugs related to scoping.

**4.Explain JavaScript const**

The “const” keyword is used to declare variables whose values should not change. Once a variable is declared using “const”, it cannot be reassigned. Like “let”, “const” is also block-scoped. It is typically used for constants or fixed references. However, if the constant holds an object or array, the contents of that object or array can still be changed, though the reference itself remains the same.

**5.Explain ES6 class fundamentals**

ES6 introduced simpler syntax for creating classes in JavaScript. Classes allow you to define blueprints for creating objects with specific properties and behaviors. They contain constructors (used to initialize values) and methos (functions) that belong to the class.

ES6 classes are easier to understand and use, especially for developers familiar with object-oriented programming.

**6.Explain ES6 class inheritance**

Class inheritance in ES6 allows one class to inherit properties and methods from another class. This is done using the “extends” keyword. It enables code reusability and helps in building relationships between classes, such as a subclass that inherits features from a parent class. Inheritance makes it easier to manage and scale complex applications by sharing common behavior across related classes.

**7.Define ES6 arrow functions**

Arrow functions are a shorter and more concise way to write functions in JavaScript. They are especially useful for small and simple functions. One of the main advantages of arrow functions is that they do not have their own “this” context, so they inherit “this” from the surrounding scope. This behavior makes them ideal for use inside callbacks and nested functions.

**8.Identify set(), map()**

* A Set is new data structure introduced in ES6 that stores unique value. It automatically removes duplicates and is useful when you want to store distinct items.
* A Map is another ES6 data structure that stores key-value pairs. Unlike regular objects, keys in Map can be of any type, and it remembers the order of insertion. Maps are often used when you need efficient lookups and ordered key-value storage.

**Codes:**

**ListOfPlayers.jsx:**

import React from 'react';

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},

];

export function **ListOfPlayers**(){

    const below70=players.**filter**(palyer=>palyer.score<=70);

    const **renderList**=(list)=>(

        <ul>

            {list.**map**((p,idx)=>(

                <li *ket*={idx}>

                    Mr. {p.name} <span>{p.score}</span>

                </li>

            ))}

        </ul>

    );

    return(

        <div>

            <h1>List of Players</h1>

            {**renderList**(players)}

            <hr/>

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

            {**renderList**(below70)}

        </div>

    );

}

**IndianPlayers.jsx:**

import React from 'react';

export function **OddPlayers**({team:[first, ,third, ,fifth]=[]}){

    return(

        <div>

            <h2>Odd Players</h2>

            <ul>

                <li>First : {first}</li>

                <li>Third : {third}</li>

                <li>Fifth : {fifth}</li>

            </ul>

        </div>

    );

}

export function **EvenPlayers**({team: [, second, , fourth, , sixth]=[]}){

    return(

        <div>

            <h2>Even Players</h2>

            <ul>

                <li>Second : {second}</li>

                <li>Fourth : {fourth}</li>

                <li>Sixth : {sixth}</li>

            </ul>

        </div>

    );

}

export function **ListOfIndianPlayers**({IndianPlayers}){

    return(

        <div>

            <h1>

                List Of Indian Players Merged:

            </h1>

            <ul>

                {IndianPlayers.**map**((name,idx)=>(

                    <li *key*={idx}>Mr. {name}</li>

                ))}

            </ul>

        </div>

    );

}

**App.js:**

import React,{useState} from 'react';

import { ListOfIndianPlayers, OddPlayers,EvenPlayers} from './components/IndianPlayers';

import { ListOfPlayers } from './components/ListOfPlayers';

function **App**() {

  const [flag,**setFlag**]=**useState**(true);

  const IndianTeam=['Sachin1','Dhoni2','Virat3','Rohit4','Raina5','Yuvaraj6'];

  const T20Players=['First Player','Second Player','Third Player'];

  const RanjiTrophyPlayers=['Fourth Player','Fifth Player','Sixth Player'];

  const IndianPlayersMerged=[...T20Players,...RanjiTrophyPlayers]

  return (

    <div *style*={{padding:'1rem'}}>

      <button *onClick*={()=>**setFlag**(!flag)}>

      Toggle Flag (Currently {flag?'True':'False'})

      </button>

      {flag?(

        <div>

          <**ListOfPlayers**/>

        </div>

      ):(

        <div>

            <h1>

              Indian Teams

            </h1>

            <**OddPlayers** *team*={IndianTeam}/>

            <hr/>

            <**EvenPlayers** *team*={IndianTeam}/>

          <hr/>

            <**ListOfIndianPlayers** *IndianPlayers*={IndianPlayersMerged}/>

        </div>

      )}

    </div>

  );

}

export default **App**;

**Output:**



