## **9. ReactJS-HOL**

## **Objectives**

* List the features of ES6
* Explain JavaScript let
* Identify the differences between var and let
* Explain JavaScript const
* Explain ES6 class fundamentals
* Explain ES6 class inheritance
* Define ES6 arrow functions
* Identify set(), map()

In this hands-on lab, you will learn how to:

* Use map() method of ES6
* Apply arrow functions of ES6
* Implement Destructuring features of ES6

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

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

1. ListofPlayers

* Declare an array with 11 players and store details of their names and scores using the map feature of ES6

A computer screen shot of a computer code

AI-generated content may be incorrect.

* Filter the players with scores below 70 using arrow functions of ES6.

A black background with white text

AI-generated content may be incorrect.

1. IndianPlayers
   1. Display the Odd Team Player and Even Team players using the Destructuring features of ES6

A screenshot of a computer program

AI-generated content may be incorrect.

* 1. Declare two arrays T20players and RanjiTrophy players and merge the two arrays and display them using the Merge feature of ES6

A black and blue text on a black background

AI-generated content may be incorrect.

Display these two components in the same home page using a simple if else in the flag variable.

**Output:**

When Flag=true

A screenshot of a computer

AI-generated content may be incorrect.

When Flag=false

A screenshot of a computer

AI-generated content may be incorrect.

**Hint:**

A screenshot of a computer screen

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.Implementation:  
A screenshot of a computer

AI-generated content may be incorrect.  
A screenshot of a computer

AI-generated content may be incorrect.

**1. Features of ES6 (ECMAScript 6)**

ES6, officially known as ECMAScript 2015, introduced a wide array of enhancements to JavaScript, making the language more modern, readable, and powerful. Some of its notable features include:

* Block-scoped variable declarations using let and const
* Arrow functions for concise function expressions
* Class-based object-oriented programming with class and extends
* Destructuring assignment for arrays and objects
* Template literals for embedded expressions
* Spread and rest operators for flexible data handling
* Native modules using import and export
* Promises for cleaner asynchronous operations
* New data structures such as Map and Set

These features significantly improve JavaScript’s ability to support scalable, modular, and readable code.

**2. Understanding let in JavaScript**

The let keyword is used to declare variables that are scoped to the block in which they are defined. Unlike var, which is function-scoped, let respects the boundaries of {} and prevents variables from leaking outside their intended scope.

let x = 5;

if (true) {

let x = 10; // This x is different from the outer x

console.log(x); // 10

}

console.log(x); // 5

Using let helps avoid bugs that can arise from variable re-declarations and unintended value changes, especially in loops and conditional blocks.

**3. Difference Between var and let**

|  |  |  |
| --- | --- | --- |
| **Feature** | **var** | **let** |
| Scope | Function-scoped | Block-scoped |
| Hoisting | Hoisted and initialized as undefined | Hoisted but not initialized |
| Redeclaration | Allowed | Not allowed in the same scope |
| Temporal Dead Zone | Not applicable | Yes (access before declaration throws error) |

In practice, let is generally preferred over var due to its stricter scoping rules and safer behavior during variable declaration.

**4. Explanation of const**

The const keyword is used to declare constants — values that cannot be reassigned after their initial assignment. Like let, const is also block-scoped. It enforces immutability at the reference level, meaning the variable must always point to the same value, though the content (in case of objects or arrays) can still be modified.

const PI = 3.14159;

PI = 3.14; // Error: Assignment to constant variable

javascript

CopyEdit

const colors = ["red", "green"];

colors.push("blue"); // Allowed

const promotes safer coding practices by preventing accidental reassignments.

**5. Fundamentals of ES6 Classes**

ES6 introduced a cleaner, more intuitive way to work with object-oriented programming in JavaScript using the class syntax. A class encapsulates data and behavior, providing a blueprint for creating objects.

class Person {

constructor(name) {

this.name = name;

}

greet() {

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

}

}

Under the hood, ES6 classes are syntactic sugar over the traditional prototype-based inheritance model of JavaScript.

**6. ES6 Class Inheritance**

Inheritance allows one class to derive from another, enabling code reuse and logical hierarchy. In ES6, inheritance is achieved using the extends keyword, and the super() function is used to call the parent class’s constructor.

class Animal {

constructor(name) {

this.name = name;

}

makeSound() {

console.log(`${this.name} makes a sound`);

}

}

class Dog extends Animal {

makeSound() {

console.log(`${this.name} barks`);

}

}

const myDog = new Dog("Tommy");

myDog.makeSound(); // Tommy barks

This pattern improves code organization, especially in large applications.

**7. Arrow Functions in ES6**

Arrow functions provide a concise syntax for writing functions. Beyond brevity, they also lexically bind the this value from their surrounding context, which makes them especially useful in callbacks and class methods.

javascript

CopyEdit

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

Arrow functions do not have their own this, arguments, or super, which helps avoid common pitfalls in JavaScript’s function scoping behavior.

**8. Introduction to Set and Map in ES6**

**Set**

A Set is a collection of unique values. It automatically removes duplicates and provides efficient methods for data operations like adding, deleting, and checking existence.

javascript

CopyEdit

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

console.log(numbers); // Set {1, 2, 3}

**Map**

A Map stores key-value pairs where keys can be of any type, including objects. It maintains the order of insertion and allows retrieval using the exact key.

const capitals = new Map();

capitals.set("France", "Paris");

capitals.set("India", "New Delhi");

console.log(capitals.get("India")); // New Delhi

These data structures offer more flexible and performant alternatives to arrays and plain objects for specific use cases.

## Objectives

* Define JSX
* Explain about ECMA Script
* Explain React.createElement()
* Explain how to create React nodes with JSX
* Define how to render JSX to DOM
* Explain how to use JavaScript expressions in JSX
* Explain how to use inline CSS in JSX

In this hands-on lab, you will learn how to:

* Use JSX syntax in React applications
* Use inline CSS in JSX

## Prerequisites

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## Notes

Estimated time to complete this lab: 60 minutes.

Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.

Create an element to display the heading of the page.

Attribute to display the image of the office space

Create an object of office to display the details like Name, Rent and Address.

Create a list of Object and loop through the office space item to display more data.

To apply Css, Display the color of the Rent in Red if it’s below 60000 and in Green if it’s above 60000.

Output:



Hint:

A black background with white text

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.

**10. ReactJS-HOL**

Objectives

* Define JSX
* Explain about ECMA Script
* Explain React.createElement()
* Explain how to create React nodes with JSX
* Define how to render JSX to DOM
* Explain how to use JavaScript expressions in JSX
* Explain how to use inline CSS in JSX

In this hands-on lab, you will learn how to:

* Use JSX syntax in React applications
* Use inline CSS in JSX

Prerequisites

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

Notes

Estimated time to complete this lab: 60 minutes.

Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.

Create an element to display the heading of the page.

Attribute to display the image of the office space

Create an object of office to display the details like Name, Rent and Address.

Create a list of Object and loop through the office space item to display more data.

To apply Css, Display the color of the Rent in Red if it’s below 60000 and in Green if it’s above 60000.

Output:



Hint:

A black background with white text

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.

**Implementation:**

**A screenshot of a computer

AI-generated content may be incorrect.**

A screenshot of a computer

AI-generated content may be incorrect.

**11. ReactJS-HOL**

## **Objectives**

* Explain React events
* Explain about event handlers
* Define Synthetic event
* Identify React event naming convention

In this hands-on lab, you will learn how to:

* Implement Event handling concept in React applications
* Use this keyword
* Use synthetic event

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **90 minutes.**

Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML.

1. Create “Increment” button to increase the value of the counter and “Decrement” button to decrease the value of the counter. The “Increase” button should invoke multiple methods.
   1. To increment the value
   2. Say Hello followed by a static message.

A screenshot of a computer

AI-generated content may be incorrect.

1. Create a button “Say Welcome” which invokes the function which takes “welcome” as an argument.

A screenshot of a computer

AI-generated content may be incorrect.

1. Create a button which invokes synthetic event “OnPress” which display “I was clicked”

A screenshot of a computer

AI-generated content may be incorrect.

Create a “CurrencyConvertor” component which will convert the Indian Rupees to Euro when the Convert button is clicked.

Handle the Click event of the button to invoke the handleSubmit event and handle the conversion of the euro to rupees.

A screenshot of a computer

AI-generated content may be incorrect.

**Implementation :**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**12. ReactJS-HOL**

## **Objectives**

* Explain about conditional rendering in React
* Define element variables
* Explain how to prevent components from rendering

In this hands-on lab, you will learn how to:

* Implement conditional rendering in React applications

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

Create a React Application named “ticketbookingapp” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.

The Login and Logout buttons should accordingly display different pages. Once the user is logged in the User page should be displayed. When the user clicks on Logout, the Guest page should be displayed.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Hint:**

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**A computer screen shot of a program code

AI-generated content may be incorrect.Implementation:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

## **Objectives**

* Explain various ways of conditional rendering
* Explain how to render multiple components
* Define list component
* Explain about keys in React applications
* Explain how to extract components with keys
* Explain React Map, map() function

In this hands-on lab, you will learn how to:

* Implement conditional rendering in React applications

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

Create a React App named “bloggerapp” in with 3 components.

1. Book Details
2. Blog Details
3. Course Details

Implement this with as many ways possible of Conditional Rendering.

A screenshot of a computer

AI-generated content may be incorrect.

**Hint:**

A computer screen with text

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A black background with text

AI-generated content may be incorrect.

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A computer screen shot of a black screen

AI-generated content may be incorrect.**