ES6 and React Hands-on Lab – Theory

Objectives:

1. List the features of ES6
2. Explain JavaScript let
3. Identify the differences between var and let
4. Explain JavaScript const
5. Explain ES6 class fundamentals
6. Explain ES6 class inheritance
7. Define ES6 arrow functions
8. Identify set(), map()

Features of ES6

ES6 (ECMAScript 2015) introduced several powerful features including let & const, arrow functions, classes, template literals, destructuring, default parameters, rest/spread operators, promises, and modules.

JavaScript let

`let` allows block-scoped variable declaration. Unlike `var`, `let` does not hoist to the top of the function and avoids redeclaration.

var vs let

- `var` is function-scoped, while `let` is block-scoped. - `var` can be redeclared and updated; `let` can be updated but not redeclared within the same scope. - `var` is hoisted with undefined; `let` is not accessible before declaration.

JavaScript const

`const` declares block-scoped constants. The value cannot be reassigned, but objects declared with `const` can have their contents mutated.

ES6 Class Fundamentals

ES6 classes are syntactic sugar over JavaScript's existing prototype-based inheritance. They support constructor functions and methods within the class body.

ES6 Class Inheritance

Using the `extends` keyword, one class can inherit from another. The `super` keyword is used to call the constructor and methods of the parent class.

Arrow Functions

Arrow functions provide a concise syntax and do not have their own `this`, making them ideal for callbacks and short functions. Example: `(a, b) => a + b`

set() and map()

- `Set` stores unique values of any type. Example: `new Set([1, 2, 2])` results in `{1, 2}`. - `Map` stores key-value pairs and remembers the insertion order. Keys can be of any type.