**REACT**

**WEEK-7 HANDSON-9 EXPLANATION**

**List the features of ECMAScript 6 (ES6)**

Introduced to JavaScript in 2015 as a significant update (also known as ECMAScript 2015)

The keyword let or const allows for declaration of variables.

Arrow functions for more concise syntax and lexical this expressions.

Backticks: Template literals for dynamic construction of strings.

Default parameters in functions.

Destructuring assignment of arrays and objects.

Spread and rest operators (...).

Object property enhancement and shorthand syntax.

Classes and inheritance class and extends.

Promissory style of asynchronous programming.

Module import and export.

Set and Map data structures.

**Explain let in JavaScript.**

Let helps to declare block-scoped variables.

The let variable is not hoisted to the top of its scope.

Reassigned in a single scope, but not redeclared.

Example:

let name = "Nikitha";

name = "Potluri"; // valid

Identify the differences between var and let.

Feature var let

Scope Function-scoped Block-scoped

Hoisting Yes, initialized as undefined Yes, but not initialized

Redeclaration Allowed Not allowed in same scope

Reassignment Allowed Allowed

**Explain JavaScript Const**

const is a form of declaring block-scoped constants.

The value assigned once cannot be changed or reassigned.

If it is an object and arrays, the reference is constant but the contents can be modified.

Example:

const age = 25;

// age = 30; // Error

const user = { name: "Nikitha" };

user.name = "Potluri"; // valid

Explain the basics of ES6 class.

class is a blueprint for creating objects

It has been introduced to support an object-oriented approach.

It initializes object properties with a constructor().

Methods are defined inside the class body without using the function keyword.

Example:

class Person {

constructor(name) {

this.name = name;

}

greet() {

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

}

}

**Explain ES6 class inheritance.**

Inheritance is achieved through the extends keyword.

The super() method invokes the constructor of the parent class.

Example:

class Student extends Person {

constructor(name, id) {

super(name);

this.id = id;

}

display() {

return `${this.name} - ${this.id}`;

}

}

**Define the ES6 arrow functions.**

Arrow functions provide a shortened way to write functions.

Do not have their own this and inherit this from the parent scope.

Syntax:

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

Identify set(), map()

Set:

A collection of unique values

Automatically removes duplicates

Provides methods like add(), has(), delete(), clear()

Example:

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

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

Map:

A collection of key-value pairs

Keeps the order of insertion

Keys can be of every type (object, function, etc.)

Example:

const myMap = new Map();

myMap.set("name", "Nikitha");

console.log(myMap.get("name")); // Nikitha