Javascript

1.Write output of below programs -

```
console.log(1+"2"+"3"); // 123

console.log("2"+1+"3"); // 213

console.log(1++); //Invalid left-hand side expression in postfix
operation

console.log(++1); //Invalid left-hand side expression in prefix
operation
```

2.check duplicate value in array -

```
const array = [1, 4, 8, 2, 4, 1, 6, 2, 9, 7];

function findDuplicates(arr) {

    return arr.filter((currentValue, currentIndex) =>
arr.indexOf(currentValue) !== currentIndex);

}

console.log(findDuplicates(array));
```

The code uses filter method to create a new array containing only the elements that have duplicates in the original array. It does so by checking whether the index of the current element is not equal to its index in the original array. If this condition is true, it means the element is a duplicate.

3. difference between let and var -

```
function varTest() {
    var x = 1;
    if (true) {
        var x = 2;  // same variable!
        console.log(x);  // 2
    }
    console.log(x);  // 2
}
varTest();
// ============
function letTest() {
    let x = 1;
    if (true) {
        let x = 2;  // different variable
        console.log(x);  // 2
    }
    console.log(x);  // 1
}
```

```
letTest();
```

4. Remove duplicates from an array and return unique values.

Ans -

```
const originalArray = [1, 2, 3, 4, 3, 2, 1];

function removeDuplicates(array) {

    const uniqueArray = [];

    for (let i = 0; i < array.length; i++) {

/*

Check for uniqueness: For each element, it checks whether it already exists in the uniqueArray using the includes method. If the element is not already in the array, it is added.

*/

        if (!uniqueArray.includes(array[i])) {

            uniqueArray.push(array[i]);

        }

    }

    return uniqueArray;

}

const uniqueArray = removeDuplicates(originalArray);

console.log(uniqueArray); // Output: [1, 2, 3, 4]
```

What is the set method used in javascript ?

Ans - set is used to return values in an array and ignore the duplicate values.

```
const ages = [26, 27, 26, 26, 28, 28, 29, 29, 30]

const uniqueAges = [...new Set(ages)]

console.log(uniqueAges) //Array [26, 27, 28, 29, 30]
```

5. What will the following code output?

```
(function() {

    var a = b = 5;

  })();

  console.log(b); // 5
```

Ans -

The code above will output 5.

6.write a program to square array elements.

Ans -

```
let arr = [1, 6, 7, 9];

let result = arr.map(x => x ** 2); // ** is es6 exponential

console.log(result); //[1,36,49,81]
```

7. checking duplicate value in array -

1st way -

```
let array1 = [5,6,7,8,13,8,6];

function dublicateFunction(arg) {

    return arg.filter( (item,i) => arg.indexOf(arg[i]) !== i);

}

console.log(dublicateFunction(array1));// [8,6]
```

8. Explain closure with an example.

A closure in JavaScript is created when a function is defined within another function and has access to the outer function's variables

Note - Closures are useful because they let you associate some data with a function that operates on that data.

```
function outerFunction() {

    // Outer function's variable

    let outerVariable = "I am from the outer function";

    // Inner function (closure)

    function innerFunction() {

      // Accessing the outer function's variable

      console.log(outerVariable);

    }
```

```
    // Returning the inner function
    return innerFunction;
  }
  // Creating a closure by invoking outerFunction
  const closure = outerFunction();
  // Invoking the closure (which still has access to outerVariable)
  closure(); // Output: I am from the outer function
```

9.Find highest number in below array.

```
const arr = [2, 4, 12, 3, 22, 10, 19, 2];

const highestNumber = Math.max(...arr);

console.log("The highest number is:", highestNumber);
```

Using for loop -

```
const arr = [2, 4, 12, 3, 22, 10, 19, 2];

// Initialize the highestNumber variable with the first element of the
array

let highestNumber = arr[0];

// Iterate over the array starting from the second element

for (let i = 1; i < arr.length; i++) {

    // Compare the current element with the current highestNumber
```

```
    if (arr[i] > highestNumber) {

        // If the current element is greater, update highestNumber

        highestNumber = arr[i];

    }

}

console.log("The highest number is:", highestNumber);
```

10. Sort below array item in ascending order.

Const numbers = [2,4,1,8,6,9]

Ans-

```
const numbers = [2, 4, 1, 8, 6, 9];

// Use the sort method with a compare function for numerical sorting

numbers.sort((a, b) => a - b);

console.log(numbers); //[1, 2, 4, 6, 8, 9]
```

React JS

1. Write a component to call below API and render data.

https://jsonplaceholder.typicode.com/posts

Ans -

Sandbox Link -
https://codesandbox.io/s/stupefied-ardinghelli-rh97n?file=/src/App.js

App.js

```
import { useEffect, useState } from "react";

import axios from "axios";

const myInfo = {

  name: "ABC",

  ctc: 77787,

  city: "Bangalore"

};

export default function App() {

  const [info, updateInfo] = useState(myInfo);

  const [dbInfo, updateDBInfo] = useState([]);

  useEffect(() => {

    const myFunction = async () => {

      const response = await axios.get(

        "https://jsonplaceholder.typicode.com/posts"

      );

      updateDBInfo(response.data);

    };

    myFunction();

  }, []);
```

```
function handleClick(e) {
  e.preventDefault();
  updateInfo({ ...info, city: "Hyderabad" });
}
return (
  <div className="App">
    <h1>{info.name}</h1>
    <h1>{info.ctc}</h1>
    <h1>{info.city}</h1>
    <button onClick={(e) => handleClick(e)}>Click me</button>
    {dbInfo &&
      dbInfo.length > 0 &&
      dbInfo.map((t) => {
        return (
          <div>
            <h2>{t.title}</h2>
            <p>{t.userId}</p>
          </div>
        );
      })}
  </div>
);
```

```
}
```

2. **Make a simple counter app (increment and decrement) using a functional component.**

Ans -

```
import React, { useState } from 'react';

function App() {

  // Declare a new state variable, which we'll call "count"

  const [count, setCount] = useState(0);

  return (

    <div>

      <p>You clicked {count} times</p>

      <button onClick={() => setCount(count + 1)}>

        Click me

      </button>

    </div>

  );

}

export default App;
```

3. **Validate name and email fields.**

   Ans -

4. **Add city and population for particular selected States.**

   Ans -

   Code Sandbox Link -

   https://codesandbox.io/s/kind-nightingale-2zp90?file=/src/App.js:0-2279

```
import React, { useState } from "react";

import "./styles.css";

import StateTable from "./StateTable";

import SampleAPIFetch from "./SampleAPIFetch";

export default function App() {

  const [stateValue, updateStateValue] = useState("");

  const [cityVale, updateCityValue] = useState("");

  const [populationCount, updatePopulation] = useState(0);

  const [karnatakaCityList, updateKarnatakaCityList] = useState([]);

  const [mahrashtraCityList, updateMaharastraCityList] =
useState([]);

  const handleSave = () => {
```

```
    if (stateValue && cityVale && populationCount) {
      let cityValues = {
        state: stateValue,
        city: cityVale,
        population: populationCount
      };
      if (stateValue === "KARNATAKA") {
        updateKarnatakaCityList([...karnatakaCityList, cityValues]);
        clearValues();
      } else {
        updateMaharastraCityList([...mahrashtraCityList,
cityValues]);
        clearValues();
      }
    }
  };
  const clearValues = () => {
    updateCityValue("");
    updatePopulation("");
  };
  return (
    <div className="App">
```

```
<div>
  <label>Select State:</label>
  <select
    onChange={(event) => {
      updateStateValue(event.target.value);
    }}
  >
    <option value="MAHARASHTRA">MAHARASHTRA</option>
    <option value="KARNATAKA">KARNATAKA</option>
  </select>
  <br></br>
  <br></br>
  <label> City: </label>
  <input
    type="text"
    placeholder="enter city name"
    value={cityVale ? cityVale : ""}
    onChange={(e) => {
      updateCityValue(e.target.value);
    }}
  />
  <br />
```

```
        <br></br>
        <label>Population: </label>
        <input
          type="text"
          value={populationCount ? populationCount : ""}
          placeholder="enter population"
          onChange={(e) => {
            updatePopulation(e.target.value);
          }}
        />
        <br />
        <br></br>
        <button onClick={handleSave}>SAVE</button>
      </div>
      <br />
      <br></br>
      <StateTable cityList={karnatakaCityList} state="KARNATAKA" />
      <StateTable cityList={mahrashtraCityList} state="MAHARASHTRA"
/>
      <SampleAPIFetch />
    </div>
  );
```

```
}
```