**NAME: DHANRAJ SUBHASH KORE**

**DIV: TY B** **BATCH: B3**

**ROLL NO: 60**

**SUB: CNT LAB ASSIGNMENT**

Q. Develop a interactive website using HTML, CSS and JavaScript to implement bubble sort and quick sort.

**Bubble Sort**

**HTML FILE :**

<!DOCTYPE html>

<html>

<head>

    <title>Bubble Sort</title>

  <meta charset="UTF-8">

    <script src= "https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.8.0/p5.min.js" type="text/javascript"></script>

    <style>

        body {

            padding: 0;

        }

        canvas {

            vertical-align: top;

        }

    </style>

</head>

<body>

    <script type="text/javascript">

        // Set Global Variables

        let values = [];

        let w = 20;

        // To store the state of each bar

        // in order to change the color

        let states = [];

        function setup() {

            // Create Canvas of Size Windows

            // Width \* Windows Height

            createCanvas(800, 400);

            // Insert Random values in array

            values = new Array(floor(width/w));

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

                values[i] = float(random(height));

                states[i] = -1;

            }

            // Print Unsorted Array

            print("Unsorted Array:" + values);

            // Call to bubble sort function

            bubbleSort(values, 0, values.length);

            // Print Sorted Array

            print("Sorted Array:" + values);

        }

        // Definition of bubble sort

        async function bubbleSort(arr, start, end) {

            if(start >= end) {

                return;

            }

            for(var i = 0; i < end-1; i++) {

                for(var j = 0; j < end-i-1; j++) {

                    if(arr[j] >= arr[j+1]) {

                        states[j] = 1;

                        // Call to swap function

                        await swap(arr, j, j+1);

                        states[j+1] = 0;

                    }

                    states[j] = 2;

                }

            }

            return arr;

        }

        // Definition of draw function

        function draw() {

            background(51);

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

                stroke(0);

                fill(255);

                if(states[i] == 0) {

                    fill(255, 0, 0);

                }

                else if (states[i] == 1) {

                    // Element currently sorting

                    fill("#58FA82");

                }

                else {

                    fill(255);

                }

                rect(i\*w, height - values[i], w, values[i]);

            }

        }

        // Definition of swap function

        async function swap(arr, a, b) {

            await sleep(20);

            let t = arr[a];

            arr[a] = arr[b];

            arr[b] = t;

        }

        // Definition of sleep function

        function sleep(ms) {

            return new Promise(resolve => setTimeout(resolve, ms));

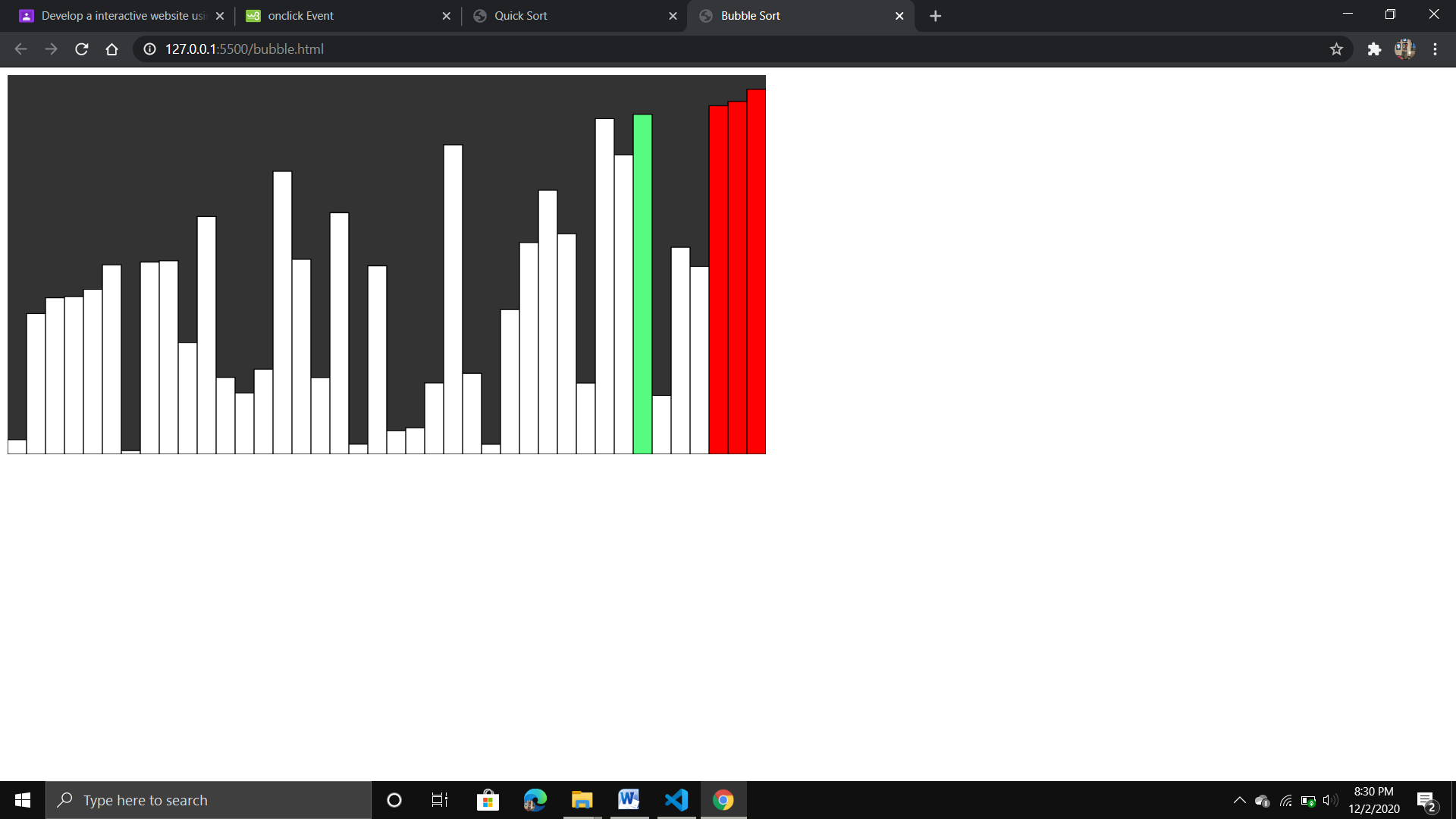
        }

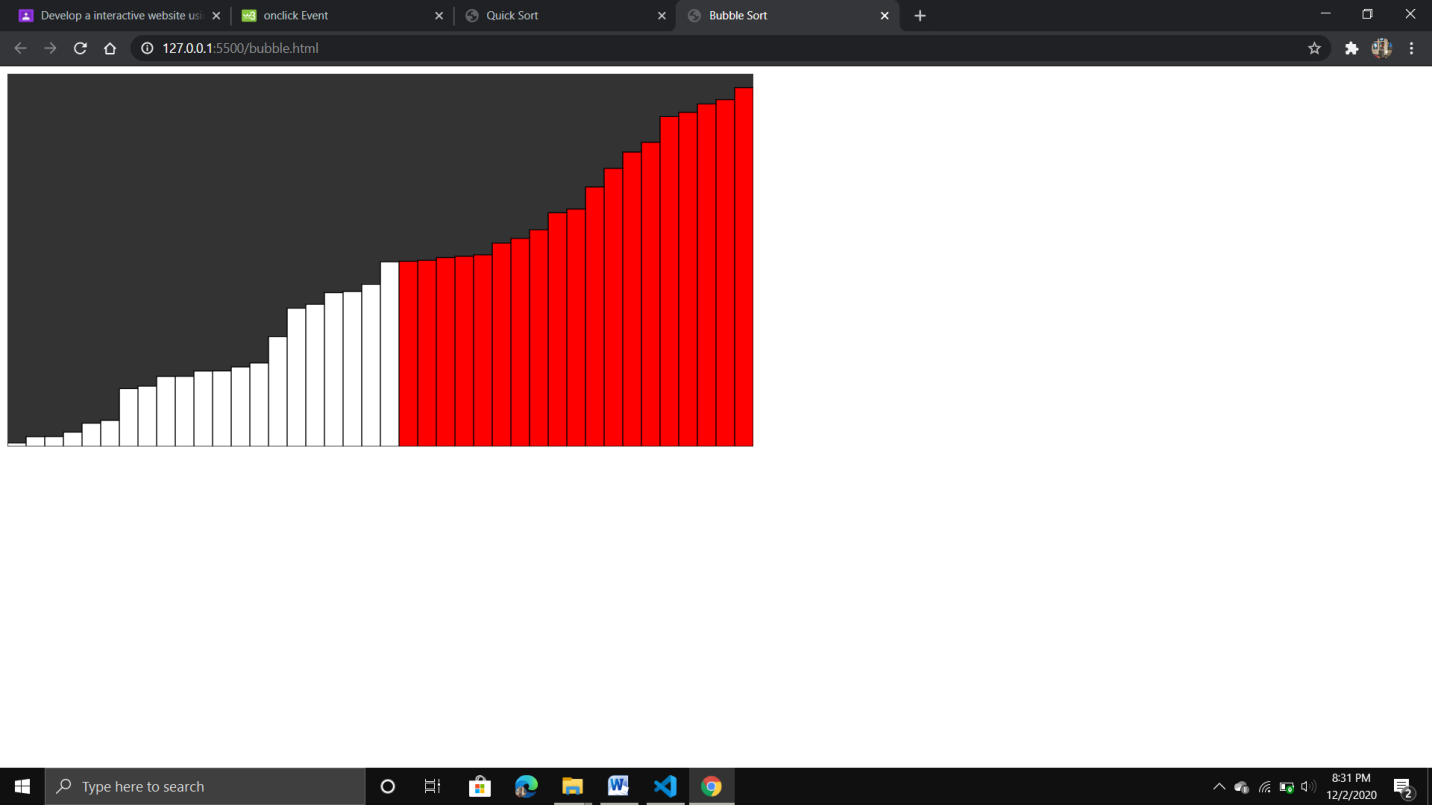
    </script>

</body>

</html>

**OUTPUT :**





**Quick Sort**

**HTML FILE :**

<!DOCTYPE html>

<html>

<head>

    <title>Quick Sort</title>

  <meta charset="UTF-8">

    <script src= "https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.8.0/p5.min.js" type="text/javascript"></script>

    <style>

        body {

            padding: 0;

        }

        canvas {

            vertical-align: top;

        }

    </style>

</head>

<body>

    <script type="text/javascript">

        // Set Global Variables

        let values = [];

        let w = 20;

        // To store the state of each bar

        // in order to change the color

        let states = [];

        function setup() {

            // Create Canvas of Size Windows

            // Width \* Windows Height

            createCanvas(800, 400);

            // Insert Random values in array

            values = new Array(floor(width/w));

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

                values[i] = float(random(height));

                states[i] = -1;

            }

            // Print Unsorted Array

            print("Unsorted Array:" + values);

            // Call to bubble sort function

            quickSort(values, 0, values.length);

            // Print Sorted Array

            print("Sorted Array:" + values);

        }

    //quick sort

    async function quickSort(arr, start, end) {

      if (start >= end) {

        return;

      }

      let index = await partition(arr, start, end);

      states[index] = -1;

      await Promise.all([

        quickSort(arr, start, index - 1),

        quickSort(arr, index + 1, end)

      ]);

      print("Sorted Array:" + values);

    }

    async function partition(arr, start, end) {

      for (let i = start; i < end; i++) {

        states[i] = 1;

      }

      let pivotValue = arr[end];

      let pivotIndex = start;

      states[pivotIndex] = 0;

      for (let i = start; i < end; i++) {

        if (arr[i] < pivotValue) {

          await swap(arr, i, pivotIndex);

          states[pivotIndex] = -1;

          pivotIndex++;

          states[pivotIndex] = 0;

        }

      }

      await swap(arr, pivotIndex, end);

      for (let i = start; i < end; i++) {

        if (i != pivotIndex) {

          states[i] = -1;

        }

      }

      return pivotIndex;

    }

        // Definition of draw function

        function draw() {

            background(51);

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

                stroke(0);

                fill(255);

                if(states[i] == 0) {

                    fill(255, 0, 0);

                }

                else if (states[i] == 1) {

                    // Element currently sorting

                    fill("#58FA82");

                }

                else {

                    fill(255);

                }

                rect(i\*w, height - values[i], w, values[i]);

            }

        }

        // Definition of swap function

        async function swap(arr, a, b) {

            await sleep(20);

            let t = arr[a];

            arr[a] = arr[b];

            arr[b] = t;

        }

        // Definition of sleep function

        function sleep(ms) {

            return new Promise(resolve => setTimeout(resolve, ms));

        }

    </script>

</body>

</html>

**OUTPUT :**

