NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES
(KARACHI CAMPUS)
Department of Computer Science
Fall 2022

Project: [Sorting Algorithms Visualized]

Group Members:

[Syed Aun Ali] - [20K-0286]

[Sufyan Imran] - [20K-0386]

Introduction:

In order for better understanding of a particular sorting algorithm, we implemented this web-based application where such sorting algorithms are visualized. It showcases the inner workings of sorting algorithms. Implemented algorithms are:

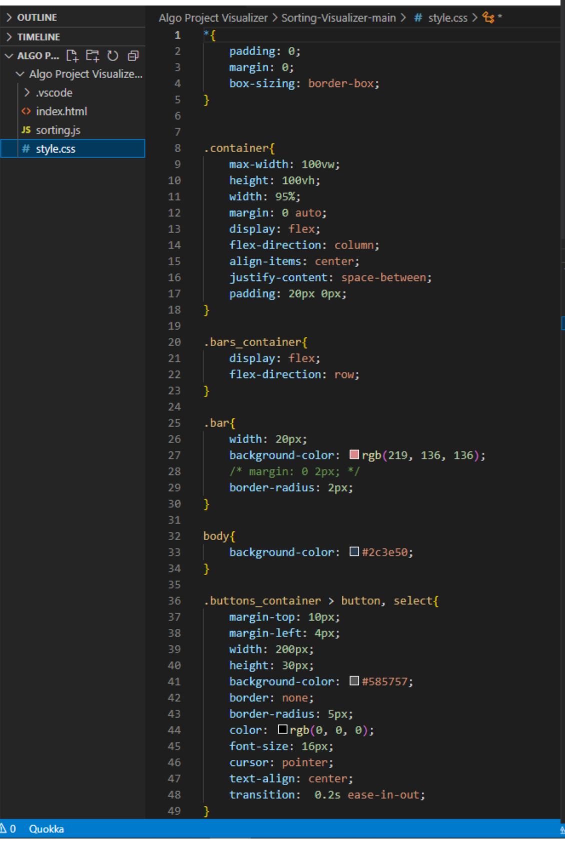
1) Bubble sort
2) Insertion sort
3) Merge sort
4) Bucket Sort
5) Count Sort
6) Radix Sort
7) Heap sort
8) Quick sort

Abstract:

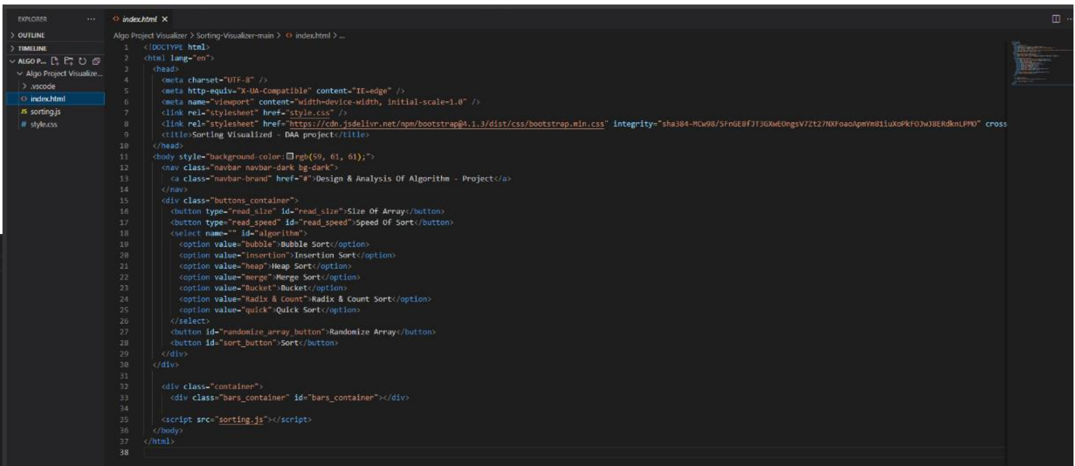
The Web application will display colored representation of steps that are being executed e.g., light-blue representing sorted array, light-red representing unsorted array, dark-red representing comparisons and pivot and finally lime-green representing iterative head/point. The application can also alter the speed of execution as well as determine the total time taken in order to sort an array (that is read from a .txt file). [Note: 'Speed Of Sort' will affect the total time taken in sorting via a chosen algorithm.]

Programming Design:

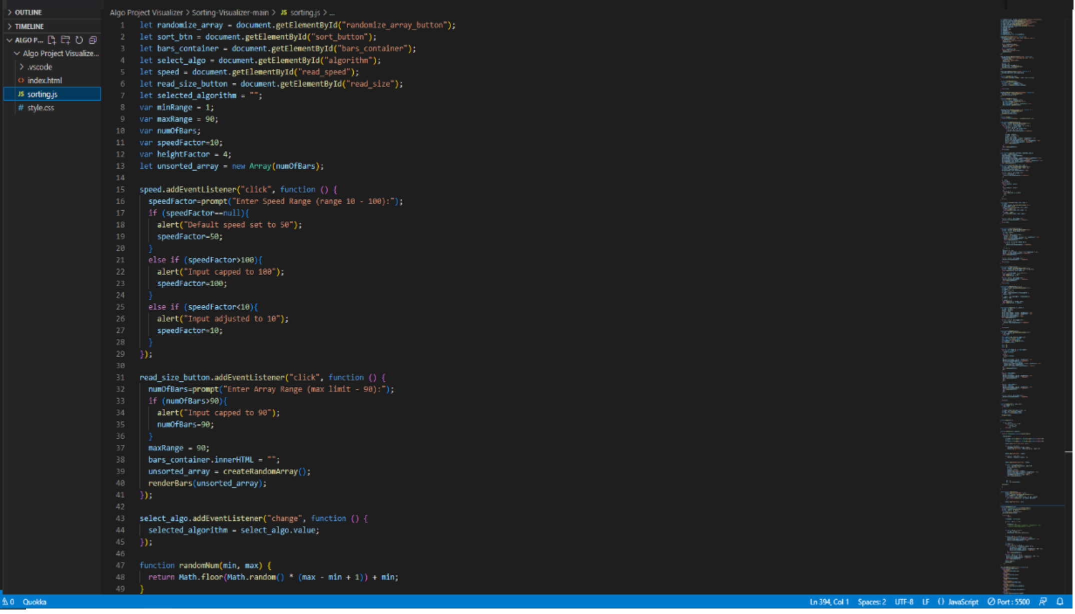
The technologies we used - HTML, CSS, JavaScript, Bootstrap.


OUTLINE
TIMELINE
ALGO P...
Algo Project Visualize...
.vscode
index.html
sorting.js
style.css
Algo Project Visualizer > Sorting-Visualizer-main > # style.css > *
Quokka

```css
*{
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}


.container{
    max-width: 100vw;
    height: 100vh;
    width: 95%;
    margin: 0 auto;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: space-between;
    padding: 20px 0px;
}

.bars_container{
    display: flex;
    flex-direction: row;
}

.bar{
    width: 20px;
    background-color: rgb(219, 136, 136);
    /* margin: 0 2px; */
    border-radius: 2px;
}

body{
    background-color: #2c3e50;
}

.buttons_container > button, select{
    margin-top: 10px;
    margin-left: 4px;
    width: 200px;
    height: 30px;
    background-color: #585757;
    border: none;
    border-radius: 5px;
    color: rgb(0, 0, 0);
    font-size: 16px;
    cursor: pointer;
    text-align: center;
    transition:  0.2s ease-in-out;
}
```


EXPLORER
index.html
OUTLINE
TIMELINE
ALGO P...
Algo Project Visualize...
.vscode
index.html
sorting.js
style.css
Algo Project Visualizer > Sorting-Visualizer-main > index.html > ...

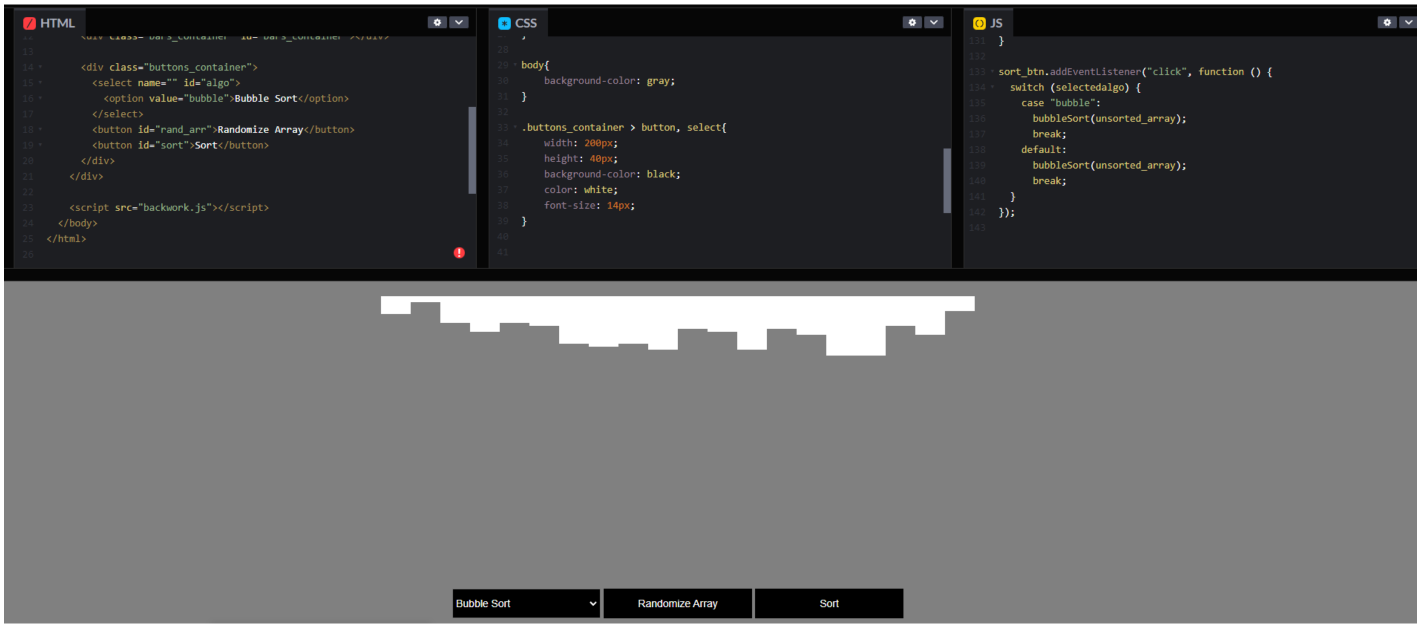
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="style.css" />
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" cross
    <title>Sorting Visualized - DAA project</title>
  </head>
  <body style="background-color: rgb(59, 61, 61);">
    <nav class="navbar navbar-dark bg-dark">
      <a class="navbar-brand" href="#">Design & Analysis Of Algorithm - Project</a>
    </nav>
    <div class="buttons_container">
      <button type="read_size" id="read_size">Size Of Array</button>
      <button type="read_speed" id="read_speed">Speed Of Sort</button>
      <select name="" id="algorithm">
        <option value="bubble">Bubble Sort</option>
        <option value="insertion">Insertion Sort</option>
        <option value="heap">Heap Sort</option>
        <option value="merge">Merge Sort</option>
        <option value="Bucket">Bucket</option>
        <option value="Radix & Count">Radix & Count Sort</option>
        <option value="quick">Quick Sort</option>
      </select>
      <button id="randomize_array_button">Randomize Array</button>
      <button id="sort_button">Sort</button>
    </div>
  </div>

    <div class="container">
      <div class="bars_container" id="bars_container"></div>

    <script src="sorting.js"></script>
  </body>
</html>

```


OUTLINE
TIMELINE
ALGO P...
Algo Project Visualize...
.vscode
index.html
sorting.js
style.css
Algo Project Visualizer > Sorting-Visualizer-main > JS sorting.js > ...
Quokka
Ln 394, Col 1   Spaces: 2   UTF-8   LF   JavaScript   Port : 5500

```javascript
let randomize_array = document.getElementById("randomize_array_button");
let sort_btn = document.getElementById("sort_button");
let bars_container = document.getElementById("bars_container");
let select_algo = document.getElementById("algorithm");
let speed = document.getElementById("read_speed");
let read_size_button = document.getElementById("read_size");
let selected_algorithm = "";
var minRange = 1;
var maxRange = 90;
var numOfBars;
var speedFactor=10;
var heightFactor = 4;
let unsorted_array = new Array(numOfBars);

speed.addEventListener("click", function () {
  speedFactor=prompt("Enter Speed Range (range 10 - 100):");
  if (speedFactor==null){
    alert("Default speed set to 50");
    speedFactor=50;
  }
  else if (speedFactor>100){
    alert("Input capped to 100");
    speedFactor=100;
  }
  else if (speedFactor<10){
    alert("Input adjusted to 10");
    speedFactor=10;
  }
});

read_size_button.addEventListener("click", function () {
  numOfBars=prompt("Enter Array Range (max limit - 90):");
  if (numOfBars>90){
    alert("Input capped to 90");
    numOfBars=90;
  }
  maxRange = 90;
  bars_container.innerHTML = "";
  unsorted_array = createRandomArray();
  renderBars(unsorted_array);
});

select_algo.addEventListener("change", function () {
  selected_algorithm = select_algo.value;
});

function randomNum(min, max) {
  return Math.floor(Math.random() * (max - min + 1)) + min;
}
```

Experimental Setup:

We used ‘codepen.io’ for our experimental setup with basic HTML, CSS and JS implementation for Bubble Sort.



HTML

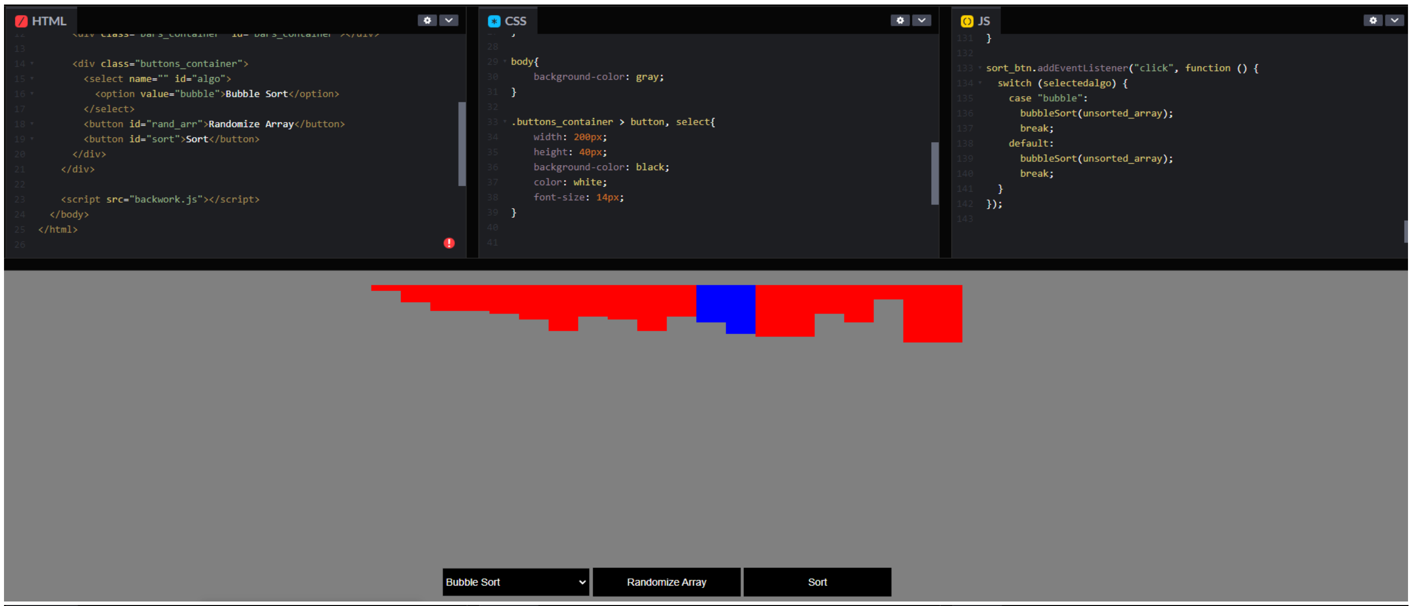
```html
      <div class="buttons_container">
        <select name="" id="algo">
          <option value="bubble">Bubble Sort</option>
        </select>
        <button id="rand_arr">Randomize Array</button>
        <button id="sort">Sort</button>
      </div>
    </div>

    <script src="backwork.js"></script>
  </body>
</html>
```

CSS

```css
body{
    background-color: gray;
}

.buttons_container > button, select{
    width: 200px;
    height: 40px;
    background-color: black;
    color: white;
    font-size: 14px;
}
```

JS

```javascript
}

sort_btn.addEventListener("click", function () {
  switch (selectedalgo) {
    case "bubble":
      bubbleSort(unsorted_array);
      break;
    default:
      bubbleSort(unsorted_array);
      break;
  }
});
```



HTML

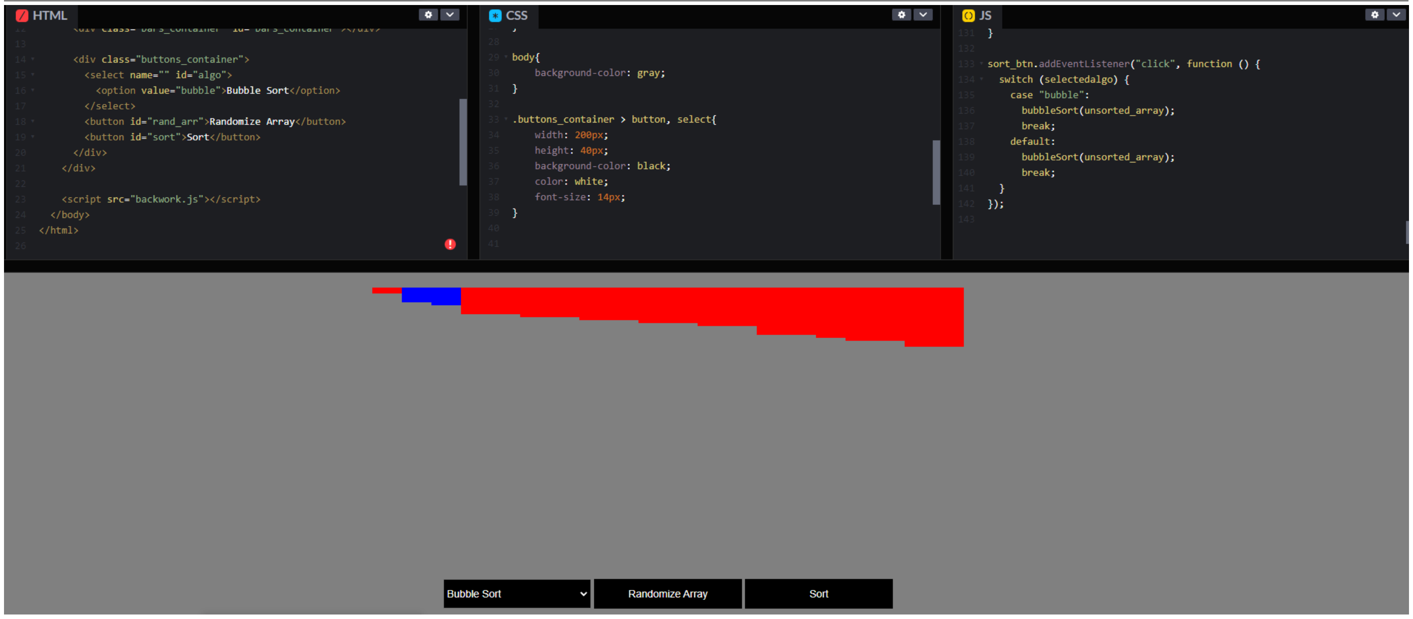
```html
      <div class="buttons_container">
        <select name="" id="algo">
          <option value="bubble">Bubble Sort</option>
        </select>
        <button id="rand_arr">Randomize Array</button>
        <button id="sort">Sort</button>
      </div>
    </div>

    <script src="backwork.js"></script>
  </body>
</html>
```

CSS

```css
body{
    background-color: gray;
}

.buttons_container > button, select{
    width: 200px;
    height: 40px;
    background-color: black;
    color: white;
    font-size: 14px;
}
```

JS

```javascript
}

sort_btn.addEventListener("click", function () {
  switch (selectedalgo) {
    case "bubble":
      bubbleSort(unsorted_array);
      break;
    default:
      bubbleSort(unsorted_array);
      break;
  }
});
```



HTML

```html
      <div class="buttons_container">
        <select name="" id="algo">
          <option value="bubble">Bubble Sort</option>
        </select>
        <button id="rand_arr">Randomize Array</button>
        <button id="sort">Sort</button>
      </div>
    </div>

    <script src="backwork.js"></script>
  </body>
</html>
```

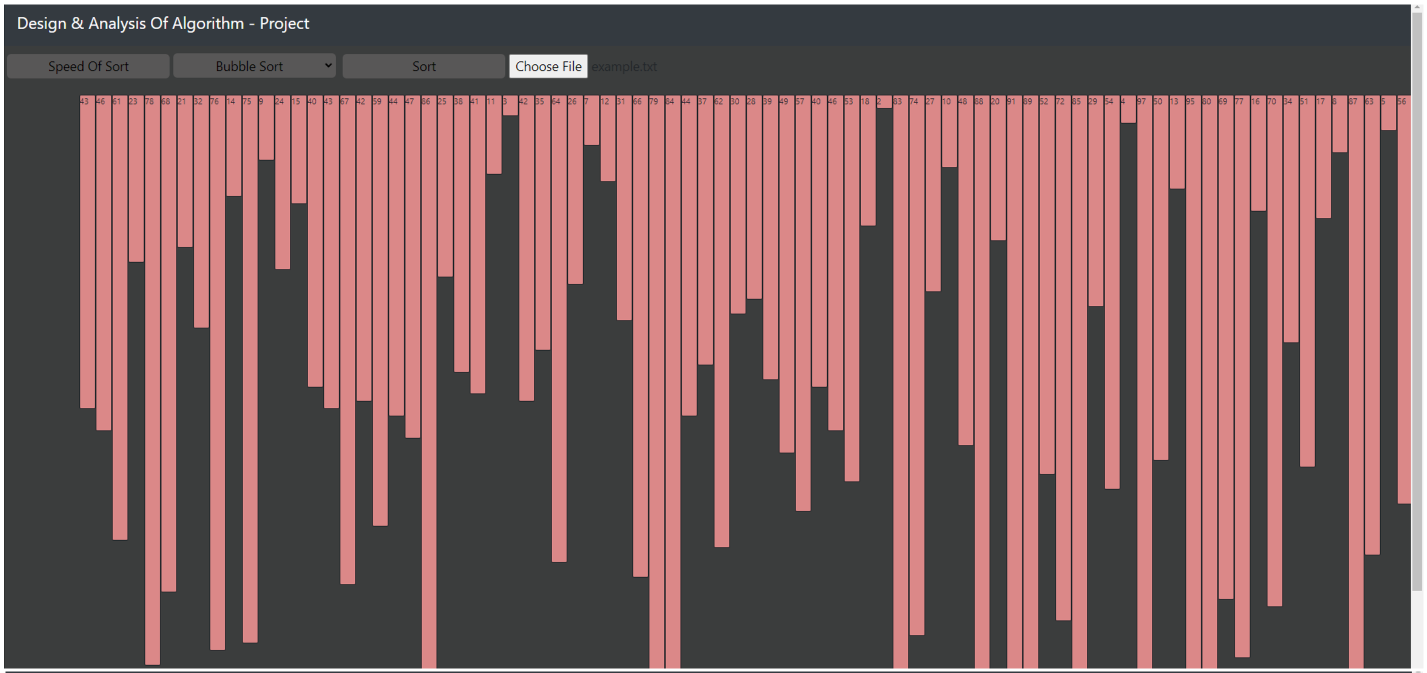
CSS

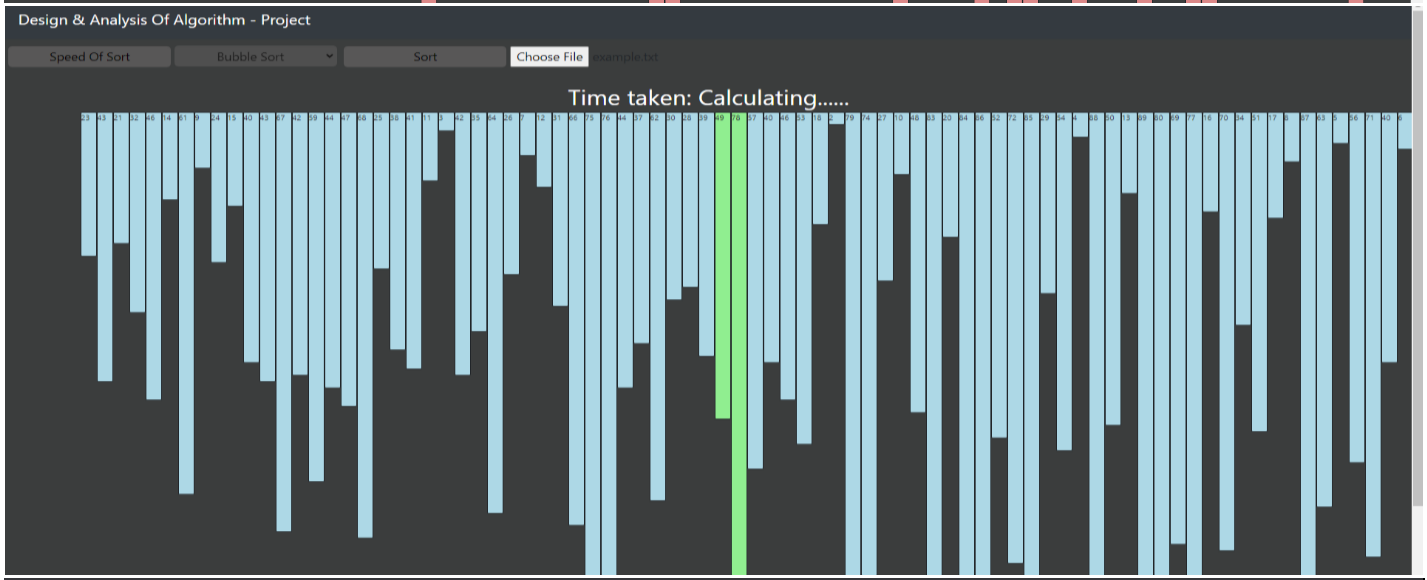
```css
body{
    background-color: gray;
}

.buttons_container > button, select{
    width: 200px;
    height: 40px;
    background-color: black;
    color: white;
    font-size: 14px;
}
```

JS

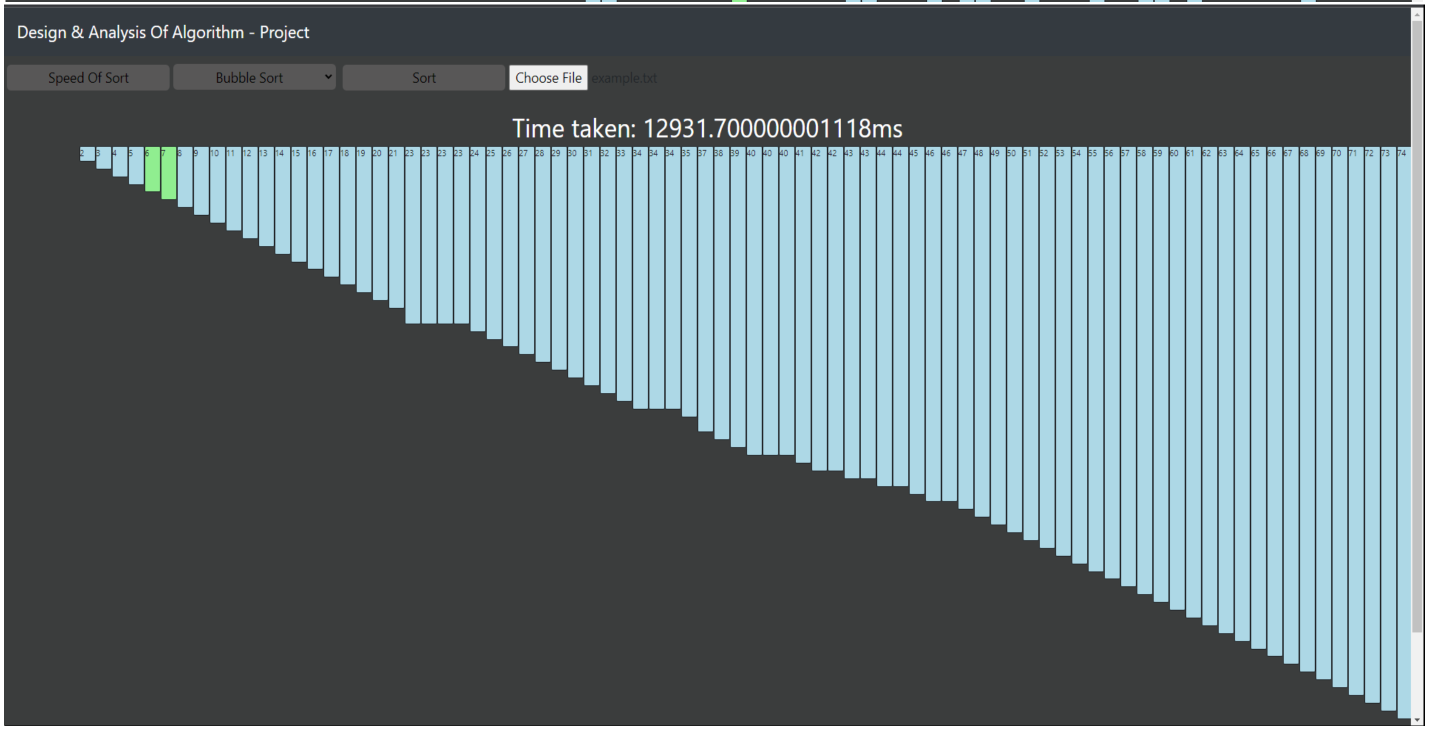
```javascript
}

sort_btn.addEventListener("click", function () {
  switch (selectedalgo) {
    case "bubble":
      bubbleSort(unsorted_array);
      break;
    default:
      bubbleSort(unsorted_array);
      break;
  }
});
```

Results and Discussion:





Time taken: Calculating......



Time taken: 12931.700000001118ms

Conclusion:

Our Web Application is accurate and efficient. It gives the required output via the required manner that is visualization of sorting algorithms.

Thank You!

References:

https://www.geeksforgeeks.org/ (for various sorting algorithms in JS)

https://www.w3schools.com/ (for front-end implementation of the website)

https://github.com/search?q=sorting+visual (for implementation reference)