Name:-Anurag Haldey

Class:-TY CC 1

Roll Number:-2215055

Enrolment Number:-MITU21BTIT0010

Experiment 4

a] Create one button on every click of button different colors should be applied

to background.

Create one button on every click of button different images should be applied.

HTML Code:-

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Color and Image Changer</title>

<style>

  body {

    margin: 0;

    display: flex;

    justify-content: center;

    align-items: center;

    min-height: 100vh;

    background-color: red; /\* Initial background color \*/

  }

  #container {

    text-align: center;

  }

  #colorButton, #imageButton {

    padding: 10px 20px;

    font-size: 16px;

    margin: 10px;

    cursor: pointer;

  }

  #backgroundImage {

    width: 300px;

    height: 300px;

    background-size: cover;

    background-position: center;

    display: flex;

    justify-content: center;

    align-items: center;

  }

</style>

</head>

<body>

<div id="container">

  <button id="colorButton">Change Color</button>

  <button id="imageButton">Change Image</button>

  <div id="backgroundImage"></div>

</div>

<script>

  const colors = ['green', 'blue', 'yellow', 'purple', 'red'];

  const images = ['Flower1.jpg', 'Flower2.jpg', 'ColdCoffee.jpg', 'Espresso.jpg', 'HotCoffee.jpg'];

  const colorButton = document.getElementById('colorButton');

  const imageButton = document.getElementById('imageButton');

  const backgroundImage = document.getElementById('backgroundImage');

  let colorIndex = 0;

  let imageIndex = 0;

  colorButton.addEventListener('click', () => {

    document.body.style.backgroundColor = colors[colorIndex];

    backgroundImage.style.backgroundImage = 'none'; // Reset background image

    colorIndex = (colorIndex + 1) % colors.length;

  });

  imageButton.addEventListener('click', () => {

    backgroundImage.style.backgroundImage = `url(${images[imageIndex]})`;

    document.body.style.backgroundColor = 'initial'; // Reset background color

    colorIndex = 0; // Reset color index when changing the image

    imageIndex = (imageIndex + 1) % images.length;

  });

</script>

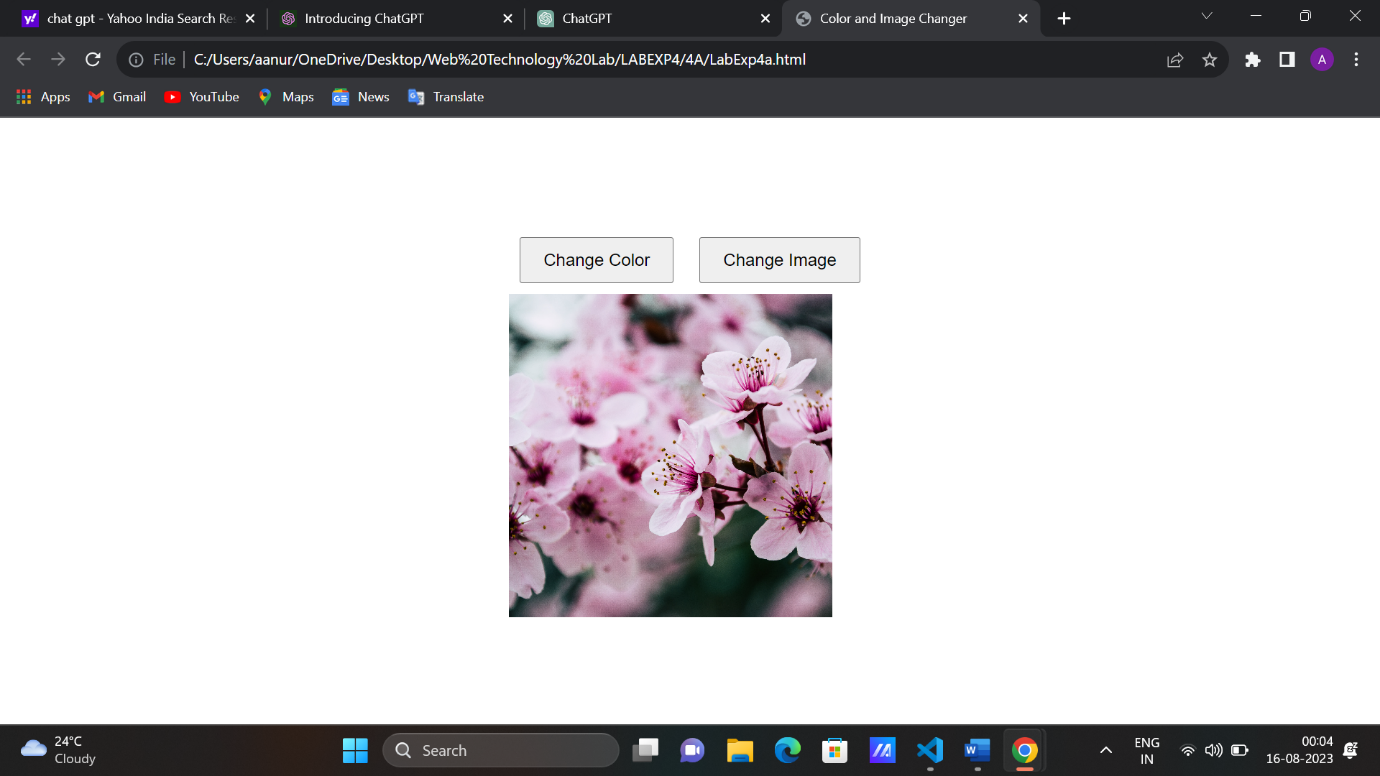
</body>

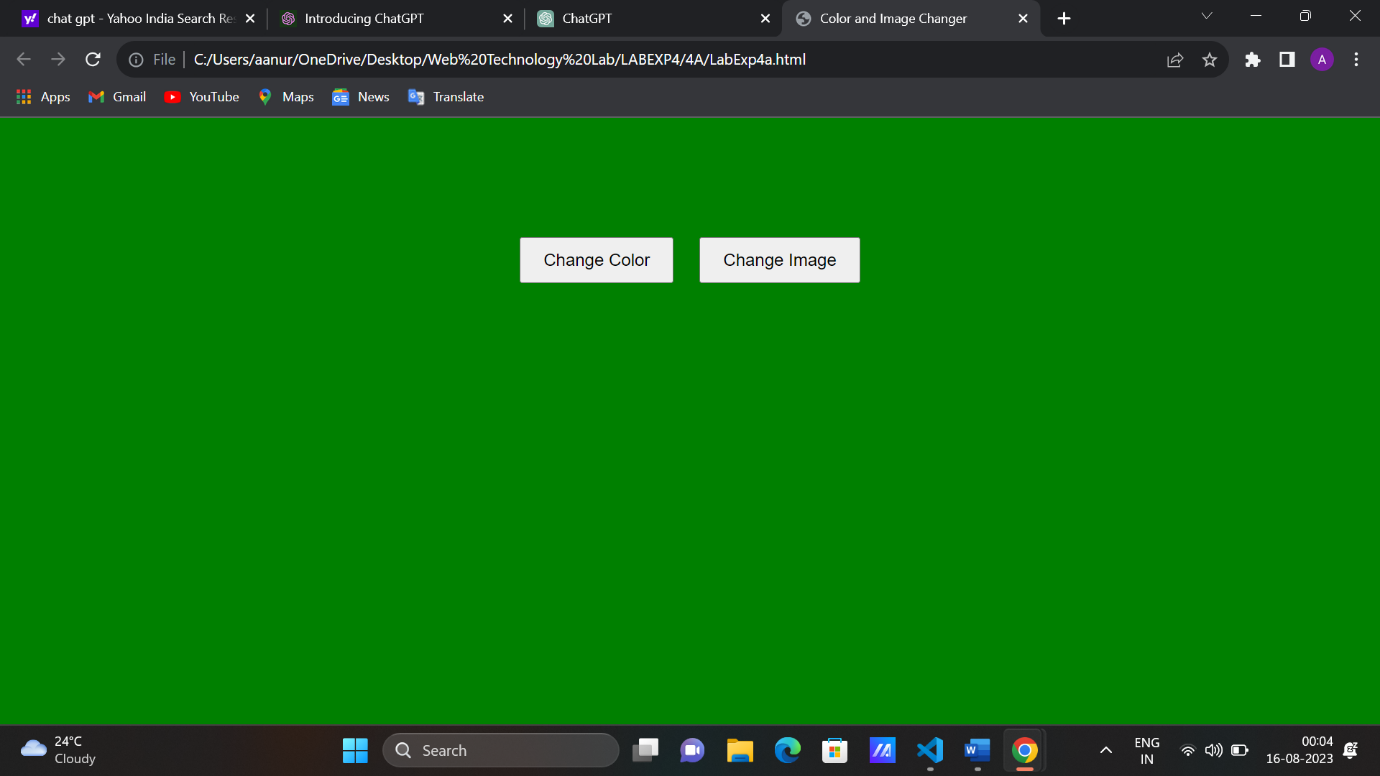
</html>

Explaination:-

* The code creates a webpage with buttons to change the background color and image.
* HTML:
  + The page structure includes a **<head>** for metadata and a **<body>** for content.
  + CSS styles are defined to center elements and set initial background color and image properties.
* JavaScript:
  + Arrays **colors** and **images** hold possible colors and image filenames.
  + Variables **colorIndex** and **imageIndex** track the current indices in the arrays.
  + Event listeners are added to the "Change Color" and "Change Image" buttons.
  + When "Change Color" button is clicked:
    - The background color of the **body** changes to the next color in **colors**.
    - The background image of **backgroundImage** is cleared.
    - The **colorIndex** is updated to loop through the colors.
  + When "Change Image" button is clicked:
    - The background image of **backgroundImage** changes to the next image in **images**.
    - The background color of the **body** is reset.
    - The **colorIndex** is reset, and the **imageIndex** is updated to loop through the images.

Output:-





b] Declare a Javascript String array of colors say colors = [“Red”, “Green”, “Blue”] Accept a value from the user and add it to the array if the value is not present in the array.

HTML Code:-

<!DOCTYPE html>

<html>

<head>

  <title>Color Array Manipulation</title>

</head>

<body>

  <div id="colorList"></div>

  <input type="text" id="newColorInput" placeholder="Enter a new color">

  <button id="addColorButton">Add Color</button>

  <script>

    // Initial array of colors

    let colors = ["Red", "Green", "Blue"];

    // Function to update the color list on the screen

    function updateColorList() {

      const colorListDiv = document.getElementById("colorList");

      colorListDiv.innerHTML = colors.map(color => `<div style="color: ${color.toLowerCase()}">${color}</div>`).join("");

    }

    // Function to check if a value is present in the array

    function isInArray(value, array) {

      return array.indexOf(value) !== -1;

    }

    // Function to add a value to the array if it's not already present

    function addColorIfNotPresent(value, array) {

      if (!isInArray(value, array)) {

        array.push(value);

        console.log(`Added '${value}' to the array.`);

        updateColorList();

      } else {

        console.log(`'${value}' is already in the array.`);

      }

    }

    const addColorButton = document.getElementById("addColorButton");

    addColorButton.addEventListener("click", () => {

      const newColorInput = document.getElementById("newColorInput");

      const userInput = newColorInput.value.trim();

      if (userInput) {

        // Convert user input to title case (e.g., "red" -> "Red")

        const formattedInput = userInput.charAt(0).toUpperCase() + userInput.slice(1).toLowerCase();

        // Add the formatted input to the array if not already present

        addColorIfNotPresent(formattedInput, colors);

        // Clear the input field

        newColorInput.value = "";

      }

    });

    // Initial update of the color list

    updateColorList();

  </script>

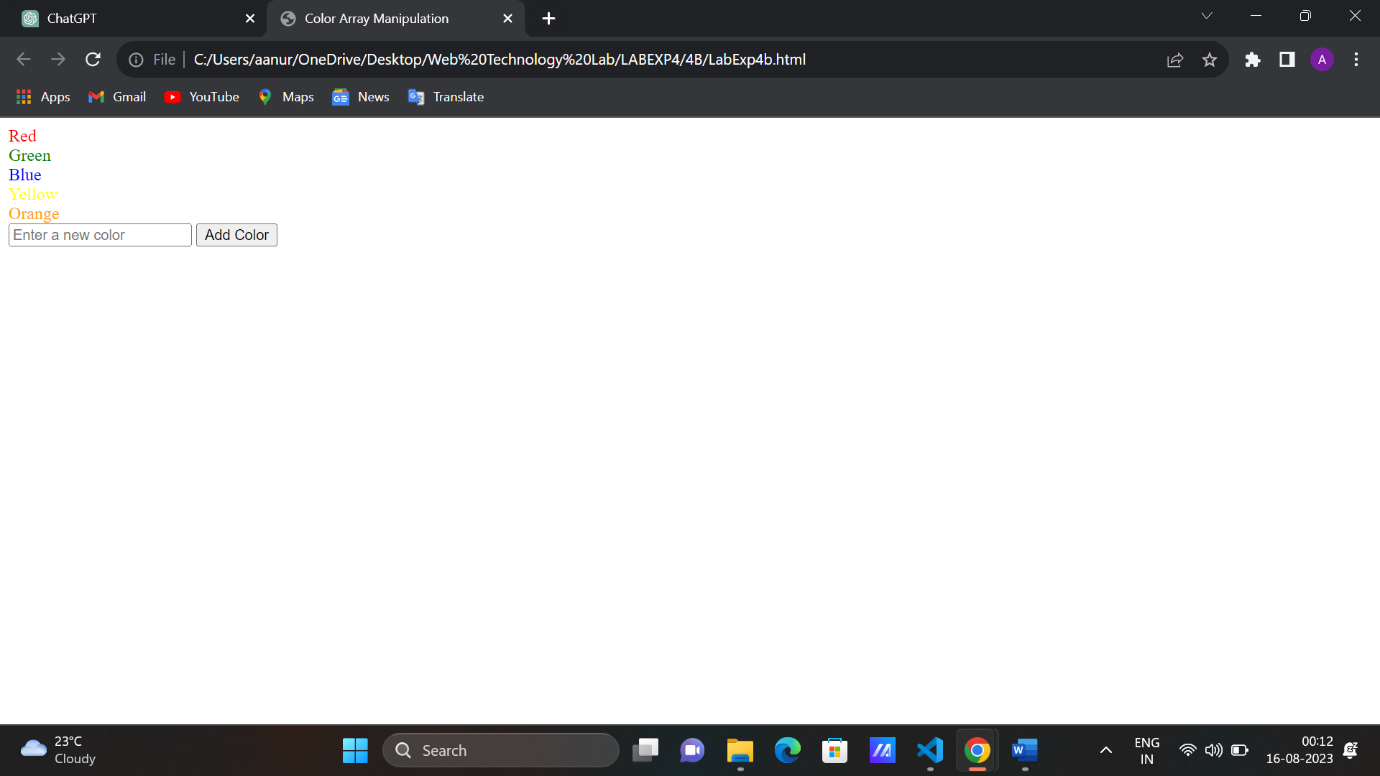
</body>

</html>

Explaination:-

* The webpage enables users to add colors to an array and displays the list of colors on the screen.
* HTML:
  + The page includes an input field for users to enter new colors, a "Add Color" button, and a **<div>** element for displaying the color list.
* JavaScript:
  + An initial array called **colors** is defined with three color names: "Red", "Green", and "Blue".
  + The **updateColorList()** function updates the color list displayed on the screen based on the **colors** array.
  + The **isInArray(value, array)** function checks if a given value is present in the array.
  + The **addColorIfNotPresent(value, array)** function adds a value to the array if it's not already present. It formats the input to title case and updates the color list on the screen.
  + An event listener is added to the "Add Color" button. When clicked, it:
    - Retrieves user input from the input field.
    - Formats the input to title case (e.g., "red" -> "Red").
    - Calls **addColorIfNotPresent()** to add the formatted input to the array if it's not already present.
    - Clears the input field.
  + The color list is initially updated using the **updateColorList()** function.

Output:-



c] Write a JavaScript code to apply font color and background color to Heading from dropdown, if font color and background color is same no changes should be reflected.

HTML Code:-

<!DOCTYPE html>

<html>

<head>

  <title>Color Styling</title>

  <style>

    body {

      display: flex;

      justify-content: center;

      align-items: center;

      min-height: 100vh;

      margin: 0;

    }

    #container {

      text-align: center;

    }

    select, h1 {

      margin: 10px;

    }

  </style>

</head>

<body>

  <div id="container">

    <select id="fontColorSelect">

      <option value="black">Black</option>

      <option value="red">Red</option>

      <option value="blue">Blue</option>

      <option value="green">Green</option>

    </select>

    <select id="bgColorSelect">

      <option value="white">White</option>

      <option value="yellow">Yellow</option>

      <option value="pink">Pink</option>

      <option value="orange">Orange</option>

    </select>

    <h1 id="heading">Sample Heading</h1>

  </div>

  <script>

    const fontColorSelect = document.getElementById("fontColorSelect");

    const bgColorSelect = document.getElementById("bgColorSelect");

    const heading = document.getElementById("heading");

    function applyStyles() {

      const selectedFontColor = fontColorSelect.value;

      const selectedBgColor = bgColorSelect.value;

      if (selectedFontColor === selectedBgColor) {

        console.log("Font color and background color are the same. No changes applied.");

        return;

      }

      heading.style.color = selectedFontColor;

      heading.style.backgroundColor = selectedBgColor;

    }

    fontColorSelect.addEventListener("change", applyStyles);

    bgColorSelect.addEventListener("change", applyStyles);

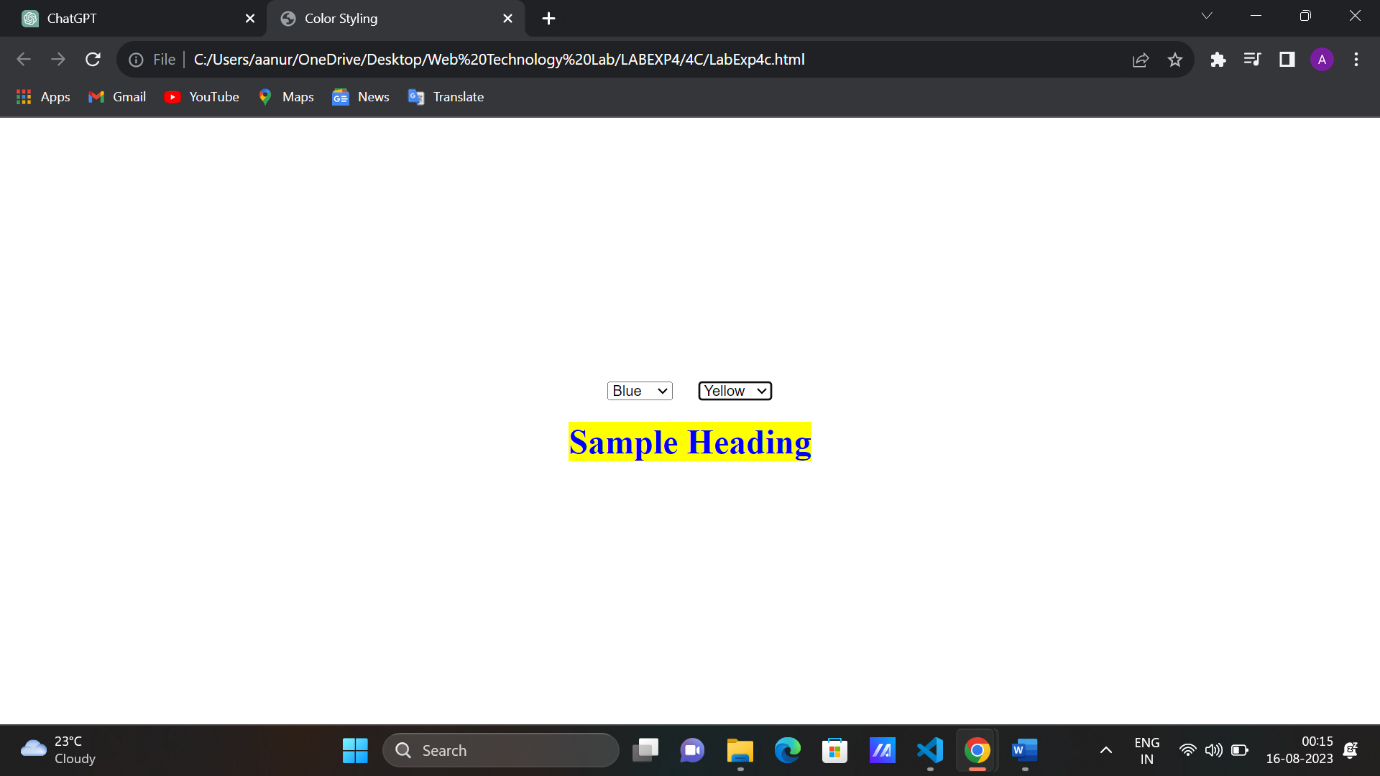
  </script>

</body>

</html>

Explaination:-

* The webpage lets users choose font and background colors from dropdowns to style a heading.
* CSS styles center the content and provide some spacing.
* JavaScript:
  + **applyStyles()** function updates heading's color and background based on dropdown selections.
  + Event listeners trigger **applyStyles()** on dropdown changes.

Output:-

d] Create an array using JavaScript and display the occurences of a specific character [For example; arr = [‘a, ‘b’, ‘a’, ‘c’, ‘z’] Output should be occurrences of a is 2].

HTML Code:-

<!DOCTYPE html>

<html>

<head>

  <title>Character Occurrences</title>

  <style>

    body {

      display: flex;

      justify-content: center;

      align-items: center;

      min-height: 100vh;

      margin: 0;

      font-family: Arial, sans-serif;

    }

    .container {

      text-align: center;

      padding: 20px;

      border: 1px solid #ccc;

      border-radius: 10px;

      background-color: #f9f9f9;

      box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);

    }

    input, button {

      padding: 10px;

      font-size: 16px;

      margin: 5px;

    }

    #output {

      margin-top: 20px;

      font-size: 18px;

      text-align: left;

    }

  </style>

</head>

<body>

  <div class="container">

    <label for="inputArray">Enter an array (comma-separated):</label>

    <input type="text" id="inputArray">

    <button id="countOccurrencesButton">Count Occurrences</button>

    <div id="output"></div>

  </div>

  <script>

    const countOccurrencesButton = document.getElementById('countOccurrencesButton');

    const inputArrayField = document.getElementById('inputArray');

    const outputDiv = document.getElementById('output');

    countOccurrencesButton.addEventListener('click', () => {

      const inputArray = inputArrayField.value.split(',').map(item => item.trim());

      const occurrences = {};

      inputArray.forEach(item => {

        if (!occurrences[item]) {

          occurrences[item] = 1;

        } else {

          occurrences[item]++;

        }

      });

      let result = '<strong>Occurrences of characters:</strong><br>';

      for (const char in occurrences) {

        result += `${char}: ${occurrences[char]}<br>`;

      }

      outputDiv.innerHTML = result;

    });

  </script>

</body>

</html>

Explaination:-

* The webpage lets users input an array of items separated by commas.
* CSS styles create a centered, bordered container with input fields and output area.
* JavaScript:
  + **countOccurrencesButton** and DOM elements are selected.
  + When the "Count Occurrences" button is clicked:
    - The input array is split and trimmed into items.
    - A loop counts occurrences of each item and stores them in the **occurrences** object.
    - The output is generated with each item's occurrences and displayed in the output area.

Output:-

