Roll no :- 48
Name :– Manav Jain

ASSIGNMENT – 5(a)

AIM:- Write a Javascript Program to study DOM Manipulation and CSS Manipulations.

LO MAPPED:- **LO4**

THEORY:-

DOM (Document Object Model) manipulation and CSS (Cascading Style Sheets) manipulation are fundamental concepts in web development that allow you to dynamically change the content and appearance of a web page using JavaScript. Let's explore each of these concepts in more detail:

1. DOM Manipulation:

The DOM is a structured representation of an HTML document. It defines the hierarchy and structure of the web page, including elements like headings, paragraphs, images, forms, and more.DOM manipulation refers to the process of interacting with and modifying the DOM using JavaScript. This allows you to add, remove, or change elements and their attributes on a web page dynamically.

Common tasks in DOM manipulation include:

- Selecting elements: You can select elements by their tag name, class, ID, or other attributes using JavaScript methods like `getElementById` & `getElementsByClassName`
- Modifying element content: You can change the text, HTML, or attributes of elements using properties like `innerHTML`, `textContent`, and `setAttribute`.

- Adding and removing elements: You can create new elements with `createElement`, append or insert elements with `appendChild`, `insertBefore`, or remove elements with `removeChild`.
- Responding to events: You can attach event listeners to elements to respond to user interactions like clicks, key presses, or form submissions.

Finding HTML Elements using following methods:

The getElementById Method: The most common way to access an HTML element is to use the id of the element. In the example above the getElementById method used id="demo" to find the element.

document.getElementsByTagName(name): Find elements by tag name

document.getElementsByClassName(name): Find elements by class name

Changing HTML Elements using following methods:

element.innerHTML = new html content: Change the inner HTML of an element

element.attribute = new value : Change the attribute value of an HTML element

element.style.property = new style: Change the style of an HTML Element using following method:

element.setAttribute(attribute, value) : Change the attribute value of an HTML element

Adding and Deleting Elements using following methods:

document.createElement(element): Create an HTML element

document.removeChild(element): Remove an HTML element

document.appendChild(element): Add an HTML element

document.replaceChild(new, old): Replace an HTML element

document.write(text): Write into the HTML output stream

2. CSS Manipulation:

CSS is a stylesheet language used to define the presentation and layout of elements on a web page. It controls aspects like colors, fonts, spacing, positioning, and more.CSS manipulation involves changing the styles of HTML elements dynamically using JavaScript.

Common tasks in CSS manipulation include:

- Changing CSS properties: You can directly modify the style properties of elements using JavaScript. For example, you can change an element's background color, font size, or margin.
- Adding and removing CSS classes: You can add or remove CSS classes to change multiple style properties at once. This is often more efficient than modifying individual properties.
- Toggle classes: You can toggle the presence of a class to switch between two styles.
- Animations and transitions: You can use JavaScript to trigger CSS animations and transitions.

Setting CSS styles using Javascript(querySelector() and querySelectorAll()):

querySelector():

• The querySelector() function takes an argument, and this argument is a string that represents the CSS selector for the element you wish to find.

• querySelector() returns the first element it finds - even if other elements exist that could get targeted by the selector.

Syntax:

var element = document.querySelector("< CSS selector >");

The querySelector() method returns the first element that matches a

CSS selector

querySelectorAll():

• The querySelectorAll() method returns all elements in the document

that matches a specified CSS selector(s), as a static NodeList object.

• The NodeList object represents a collection of nodes.

- The nodes can be accessed by index numbers.
- The index starts at 0

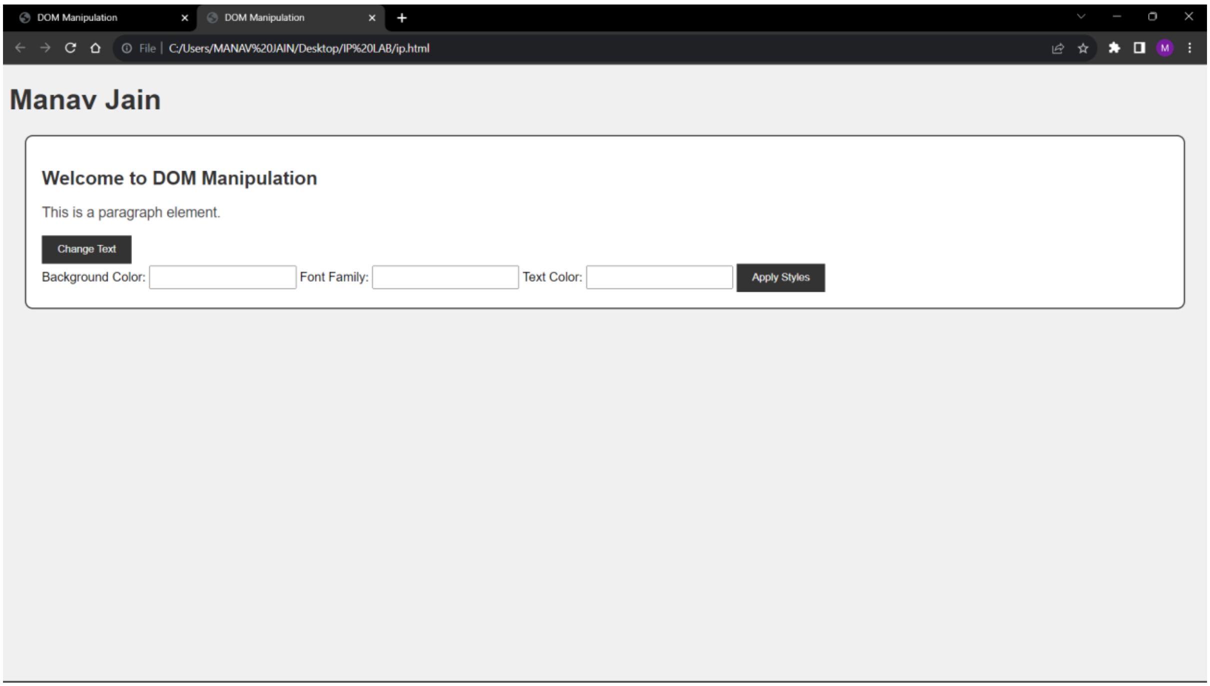
OUTPUT WITH CODE:-

```
<!DOCTYPE html>
<html>
<head>
  <title>DOM Manipulation</title>
  <style>
    /* CSS styles */
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
    }

    #container {
      margin: 20px;
      padding: 20px;
      background-color: #fff;
      border: 2px solid #333;
      border-radius: 10px;
    }

    h2 {
      font-size: 24px;
      color: #333;
    }

    p {
      font-size: 18px;
      color: #444;
    }

    button {
      padding: 10px 20px;
      background-color: #333;
```

```
          color: white;
          border: none;
          cursor: pointer;
        }

        input[type="text"] {
          padding: 5px;
          font-size: 14px;
        }

        #name {
          font-size: 36px;
          font-weight: bold;
          color: #333;
        }

        .changed-text {
          color: red; /* Change the color for changed text */
        }
      </style>
    </head>
    <body>
      <h1 id="name">Manav Jain</h1>
      <div id="container">
        <h2>Welcome to DOM Manipulation</h2>
        <p id="paragraph">This is a paragraph element.</p>
        <button id="changeTextBtn">Change Text</button>
        <br>
        <label for="backgroundColor">Background Color:</label>
        <input type="text" id="backgroundColor">
        <label for="fontFamily">Font Family:</label>
        <input type="text" id="fontFamily">
        <label for="textColor">Text Color:</label>
        <input type="text" id="textColor">
        <button id="applyStylesBtn">Apply Styles</button>
      </div>
    <script>
        // Function to change the text content of the paragraph and apply styles
        function changeText() {
            const newContent = "Text has been changed!";
            const paragraph = document.querySelector("#paragraph");
            paragraph.textContent = newContent;
            paragraph.classList.add("changed-text"); // Apply the changed-text class
        }

        // Function to apply styles to the paragraph
        function applyStyles() {
            const backgroundColor =
document.querySelector("#backgroundColor").value;
            const fontFamily = document.querySelector("#fontFamily").value;
            const textColor = document.querySelector("#textColor").value;

            const paragraph = document.querySelector("#paragraph");
            paragraph.style.backgroundColor = backgroundColor;
            paragraph.style.fontFamily = fontFamily;
            paragraph.style.color = textColor;
        }

        // Function to create and append a new element
        function appendNewElement() {
            const newDiv = document.createElement("div");
            newDiv.textContent = "This is a new div element.";

            const container = document.querySelector("#container");
            container.appendChild(newDiv);
        }

        // Add event listeners
        document.querySelector("#changeTextBtn").addEventListener("click",
changeText);
        document.querySelector("#applyStylesBtn").addEventListener("click",
applyStyles);
```

```
        document.querySelector("#container").addEventListener("click",
appendNewElement);
    </script>
</body>
</html>
```

Output:



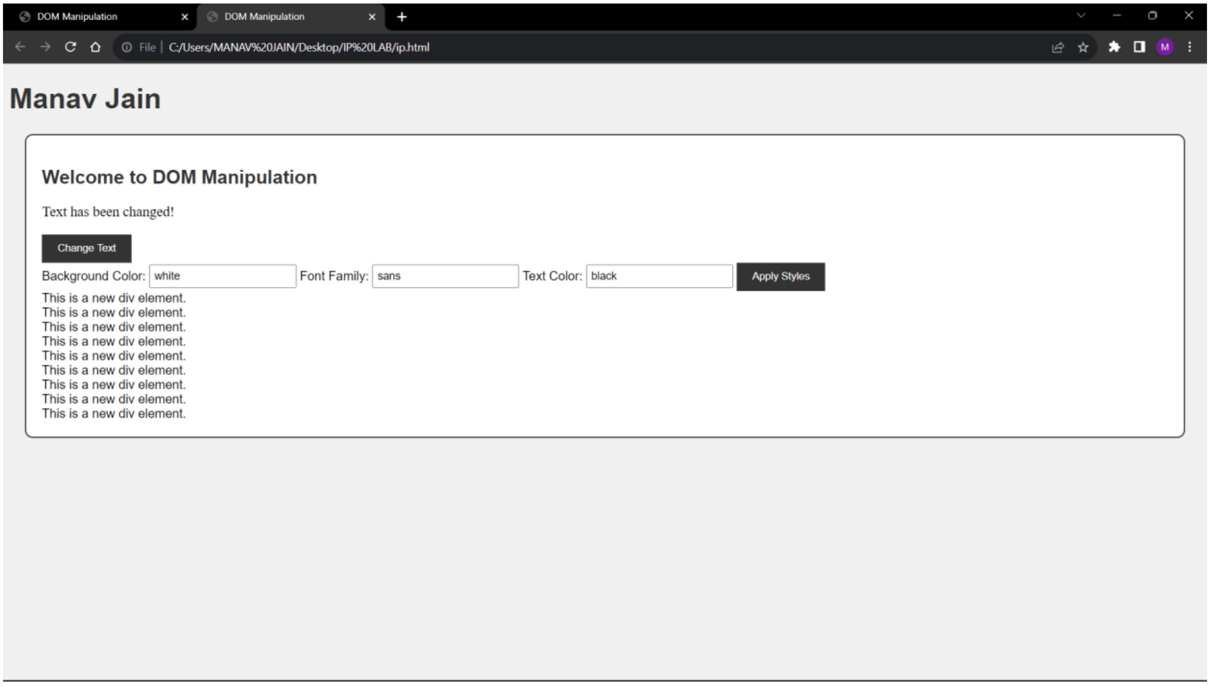
Manav Jain

Welcome to DOM Manipulation

This is a paragraph element.

Change Text

Background Color: Font Family: Text Color: Apply Styles



Manav Jain

Welcome to DOM Manipulation

Text has been changed!

Change Text

Background Color: white Font Family: sans Text Color: black Apply Styles

```
This is a new div element.
This is a new div element.
This is a new div element.
This is a new div element.
This is a new div element.
This is a new div element.
This is a new div element.
This is a new div element.
This is a new div element.
```

<u>**CONCLUSION:-**</u> In this code, methods such as createElement and appendChild are used to create new elements and append them to the DOM. Additionally,

methods like getElementById and querySelector are used to modify the text content, effectively demonstrating the concepts of DOM and CSS manipulation.