St Francis Institute of Technology

(Autonomous Institute)

Department of Artificial Intelligence and Machine Learning

Second Year AIML Engineering (SEM-IV A.Y. 2025-26)

Web Programming Lab. Experiment Report

Experiment 5: Perform DOM Manipulation by Developing a Dynamic To-Do List Application Using JavaScript

---

1. AIM

   To perform DOM manipulation using JavaScript by developing a dynamic To-Do List application that allows users to add, display, and remove tasks interactively.

2. LAB OBJECTIVE
   a. To understand the concept of Document Object Model (DOM)
   b. To access and manipulate HTML elements using JavaScript
   c. To dynamically create, modify, and remove DOM elements
   d. To handle user actions using event listeners
   e. To develop a simple interactive web application

3. LAB OUTCOME

   After completing this experiment, the student will be able to:
   a. Explain the structure and role of the DOM
   b. Select HTML elements using JavaScript
   c. Dynamically add and remove elements from the webpage
   d. Implement event-driven logic using JavaScript
   e. Build a basic interactive web application

4. PREREQUISITE
   a. Knowledge of HTML forms and elements
   b. Understanding of JavaScript basics and events
   c. Completion of Experiment-3 (JavaScript Interactivity)

5. THEORY
   a. Document Object Model (DOM)

      The DOM is a tree-like representation of an HTML document. JavaScript uses the DOM to access and manipulate webpage elements dynamically.
      i. DOM stands for Document Object Model
      ii. It represents the HTML page as a tree of elements
      iii. JavaScript uses the DOM to access and modify webpage content

b. DOM Selection Methods

Common methods used:

i. document.getElementById( )
ii. document.querySelector( )
iii. document.querySelectorAll( )

c. DOM Manipulation

JavaScript can:

i. Create new elements
ii. Change content
iii. Modify styles
iv. Remove elements

d. Event Handling

Events such as button clicks trigger JavaScript functions to perform actions like adding or deleting tasks.

i. What is an Event?

An **event** is an action that occurs in a web browser, such as:

➔ Clicking a button
➔ Typing in an input box
➔ Submitting a form
➔ Loading a webpage

ii. How Event Handling Works

➔ A user performs an action (example: clicks a button)
➔ The browser detects the event
➔ JavaScript executes a function linked to that event
➔ The webpage updates dynamically

iii. Two common JavaScript events are:

➔ **click event**
Occurs when a user clicks on an element such as a button or link.
➔ **keydown event**
Occurs when a user presses a key on the keyboard.

6. PROCEDURE

a. Create an HTML structure for the To-Do List.
b. Add an input field and a button for task entry.
c. Link an external JavaScript file to the HTML page.
d. Use JavaScript to read input values.
e. Dynamically create list items using DOM methods.
f. Append tasks to the list.
g. Add functionality to remove completed tasks.

h. Test the application by adding and deleting tasks.

7. RESULTS / OUTCOME EXPECTED
    a. Tasks can be added dynamically
    b. Tasks can be removed from the list
    c. Webpage updates without refresh
    d. Smooth user interaction using JavaScript

8. CONCLUSION
    a. How does DOM manipulation help in creating a dynamic To-Do List?
    b. Which DOM methods were used to add new tasks?
    c. How are events used to manage user actions?
    d. Why is JavaScript essential for dynamic web applications?
    e. How does this experiment demonstrate real-time webpage updates?

9. POST-EXPERIMENT QUESTIONS
    a. What is the DOM?
    b. Difference between innerText and innerHTML?
    c. What is appendChild() used for?
    d. How does event-driven programming work?
    e. How can DOM manipulation improve user experience?

10. REFERENCES

MDN Web Docs – DOM
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

W3Schools – JavaScript DOM
https://www.w3schools.com/js/js_htmldom.asp

JavaScript.info – DOM Manipulation
https://javascript.info/document

CODE

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>To-Do List</title>
</head>
<body>
<h2>To-Do List</h2>
<input type="text" id="taskInput" placeholder="Enter task">
<button id="addBtn">Add Task</button>
<ul id="taskList"></ul>
<script src="5.js"></script>
</body>
</html>
```

JS

```
document.getElementById("addBtn").addEventListener("click", function() {
    let taskText = document.getElementById("taskInput").value;
    if (taskText !== "") {
        let li = document.createElement("li");
        li.textContent = taskText;
        let removeBtn = document.createElement("button");
        removeBtn.textContent = "Remove";
        removeBtn.addEventListener("click", function() {
            li.remove();
        });
        li.appendChild(removeBtn);
        document.getElementById("taskList").appendChild(li);
        document.getElementById("taskInput").value = "";
    }
});
```

OUTPUT