Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment Title: JS DOM Events: Write all the tasks inside the index.js and an index.htmlfile. For each task, you should create HTML template. Template for each task should be available in index.html file (each task has a special block, that separated by the comment). For providing styles, use styles.css file. (the same situation as for HTML template, each task has its own style block).

Aim/Objective

To implement and demonstrate various JavaScript DOM event handling tasks by organizing all JavaScript logic inside index.js, corresponding HTML templates in index.html, and styling in styles.css, with clear separation for each task using comment blocks for better modularity and readability.

Description:

This activity focuses on learning and applying JavaScript DOM events by handling different tasks in a structured format. Each task includes a dedicated HTML section, corresponding JavaScript logic, and specific CSS styling, all separated clearly using comment blocks in their respective files. This approach promotes clean code organization and modular development.

Prerequisites:

- Basic understanding of HTML, CSS, and JavaScript.
- Familiarity with DOM (Document Object Model) concepts and structure.
- Knowledge of common JavaScript events (e.g., click, input, mouseover, keydown).
- Ability to link external CSS and JS files to an HTML document.
- Experience using browser developer tools for debugging and inspecting elements.

Pre-Lab: Before starting the lab activity, students should:

- Review the structure of an HTML document, including elements, attributes, and nesting.
- Understand the concept of the DOM and how JavaScript can interact with HTML elements dynamically.
- Revise basic JavaScript syntax, especially functions, variables, and conditional statements.
- Learn common DOM methods such as getElementById, querySelector, addEventListener, and innerHTML.
- Practice identifying and describing different types of events (e.g., mouse, keyboard, form events).
- Set up a basic project structure with index.html, styles.css, and index.js files linked properly.

In-Lab: During the lab session, students will:

- Create separate HTML blocks in index.html for each DOM event task, clearly marked with comment tags (e.g., <!-- Task 1 Start --> to <!-- Task 1 End -->).
- Write corresponding JavaScript code for each task in index.js, ensuring event listeners are used appropriately (e.g., click, mouseover, keyup).
- Apply specific CSS styles for each task in styles.css, separated by comments for clarity (e.g., /* Task 1 Styles */).
- Use console logging or on-screen messages to verify that the correct events are triggered.
- Debug issues using browser developer tools and inspect elements to ensure proper event binding.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Sample viva voce questions:

1. **What is the DOM, and how does JavaScript interact with it?**

2. **What is the purpose of the addEventListener() method in JavaScript?**

3. **Explain the difference between inline event handling and using external JavaScript files.**

4. **Can you name at least three commonly used DOM events and their use cases?**

5. **How do you prevent default behavior of an event (e.g., form submission)? Explain with an example.**

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date