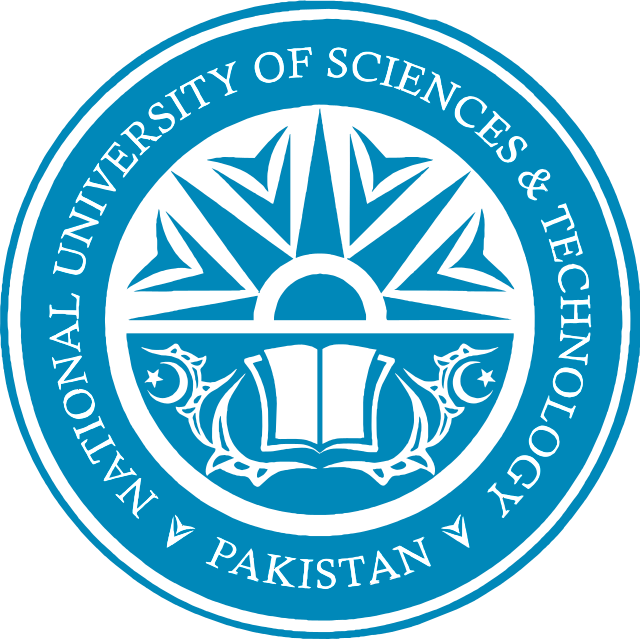


Assignment 02

**Web** Engineering

****

|  |  |
| --- | --- |
| Instructor | Sir Qasir |
| Student Name | M Anas Hassaan |
| CMSID | 410555 |
| Department | Computer Science |
| Semester | 7th |

**Q1. Explain the concept of the Document Object Model (DOM) in JavaScript. Describe how JavaScript can be used to access, modify, add, and delete elements in the DOM. Support your answer with a real-world example where DOM manipulation is used to update webpage content dynamically.**

The Document Object Model (DOM) is a critical concept in web development that provides a structured representation of an HTML or XML document. It organizes the document as a tree-like hierarchy of nodes, where each node represents an element, attribute, or text within the page. This structure allows programming languages, particularly JavaScript, to interact directly with the content, structure, and styling of a webpage. By leveraging the DOM, developers can create webpages that are not only static but also highly dynamic and responsive to user interactions.

JavaScript provides powerful tools to access elements within the DOM. Elements can be selected based on their unique identifiers, class names, tag names, or more complex CSS-style selectors. Once accessed, these elements can be modified dynamically, allowing developers to update text, change attributes like images or links, and alter visual styles such as colors, fonts, or visibility. This capability enables webpages to respond to user actions in real time, enhancing interactivity and user experience.

Beyond modification, the DOM allows developers to add or remove elements programmatically. New elements can be created and inserted into the document at specific locations, allowing content to be expanded or updated dynamically. Similarly, existing elements can be removed from the DOM, supporting functionality such as deleting items from lists, hiding notifications, or updating sections of a webpage without requiring a page reload.

A practical example of DOM manipulation is seen in task management applications. When a user adds a new task, JavaScript can dynamically create a new task element and insert it into the task list on the webpage. Likewise, tasks can be removed instantly when marked as completed. This interaction occurs seamlessly without reloading the page, demonstrating how DOM manipulation enables webpages to be dynamic, interactive, and user-friendly. Such capabilities are fundamental to modern web applications, where responsiveness and real-time content updates are essential.