Theoretical Explanation of a To-Do List Application Using JavaScript, HTML, and CSS

1. Introduction

A **To-Do List Application** is a simple project that allows users to manage tasks by adding, marking them as completed, and deleting them. This application demonstrates fundamental concepts in **HTML, CSS, and JavaScript**.

---

2. Components of the Application

The application consists of three main components:

- **HTML (Structure)**: Defines the layout of the application.
- **CSS (Styling)**: Enhances the appearance of the application.
- **JavaScript (Functionality)**: Implements logic for adding, deleting, and marking tasks as completed.

---

3. HTML Structure

The **HTML document** serves as the skeleton of the application. It consists of:

- A heading (<h2>) displaying the title **"To-Do List"**.
- An **input field** (<input>) where users can type a task.
- A **button** (<button>) labeled **"Add Task"** to add a new task.
- An **unordered list** (<ul>) that dynamically displays the tasks.

---

4. CSS Styling

The **CSS file** is used to enhance the visual appeal of the application. Key styling elements include:

- Centering the **To-Do List** in the middle of the screen.
- Adding **spacing, colors, and borders** to improve readability.
- Using **text-decoration (strikethrough effect)** to indicate completed tasks.

---

5. JavaScript Functionality

JavaScript is responsible for handling the interactivity of the application. The key functions include:

1. **Adding a Task:**
    - Retrieves the text from the input field.
    - Creates a new list item (<li>) and appends it to the task list.
    - Clears the input field after adding the task.
2. **Marking a Task as Completed:**
    - Adds an event listener to checkboxes.
    - When checked, applies a strikethrough effect on the task text.
3. **Deleting a Task:**
    - Creates a delete button for each task.
    - Clicking the button removes the corresponding task from the list.
4. **(Optional) Storing Tasks Using Local Storage:**
    - Saves tasks in the browser's local storage so they persist even after refreshing the page.

---

6. Workflow of the Application

1. **User enters a task** in the input field and clicks "Add Task."
2. **The task appears** in the task list.
3. **User marks the task as completed** by checking the checkbox.
4. **User deletes the task** using the delete button.

---

7. Conclusion

This project provides a **practical understanding of JavaScript** for manipulating the **DOM (Document Object Model)**. It helps learners build interactive web applications and enhances problem-solving skills in web development.