TITLE: CodTech IT Solutions Internship - Task Documentation: “To-DO LIST” Using CSS, HTML, JAVASCRIPT.

INTERN INFORMATION:

Name: Krupali Ravrani

ID: ICOD6629

INTRODUCTION

A To-Do List app is a fundamental tool for organizing tasks and managing daily activities efficiently. By combining HTML, CSS, and JavaScript, developers can create an intuitive and user-friendly interface that enables users to add, edit, and remove tasks seamlessly. HTML provides the structure of the app, defining the layout and elements such as input fields and buttons. CSS is utilized to style the app, ensuring a visually appealing presentation and enhancing user experience. JavaScript adds interactivity to the app, enabling dynamic updates and functionality such as task completion and sorting. Together, these technologies empower the creation of a versatile and accessible To-Do List app that simplifies task management for users.

Implementation

- Design the basic structure of the app using HTML. Define elements such as input fields for adding tasks, buttons for interaction and a container to display the list of tasks.
- Apply CSS styles to enhance the visual appeal and usability of the app. This includes defining fonts, colors, spacing, and layout to make the app aesthetically pleasing and user-friendly.
- Implement responsive design techniques using CSS to ensure the app looks and functions well across different devices and screen sizes. Consider using media queries to adjust the layout and styling for smaller screens such as mobile devices.

CODE EXPLAINATION

HTML Structure:

<!DOCTYPE html>: Defines the document type and version of HTML being used.

<head>: Contains metadata about the document, including character encoding, viewport settings, and the document title.

<link rel="stylesheet2" href="styles.css">: Links to an external CSS file for styling the To-Do List app.

: Creates a container to hold all elements of the To-Do List app

<script src="script2.js"></script>: Links to an external JavaScript file for implementing the logic of the To-Do List app.

CSS Styling:

- **Readable Fonts:** Choose readable fonts for the task list and other elements. Sans-serif fonts like Arial or Roboto are often preferred for their clarity.
- **Distinctive Task Items:** Style task items (li elements) to stand out from the rest of the content. You can use a different background color, border, or font weight to make them visually distinct.
- **Hover Effects:** Add hover effects to task items to provide visual feedback when users interact with them. For example, change the background color or add an underline effect when hovering over a task.
- **Checkbox Styling:** If your To-Do List includes checkboxes for marking tasks as complete, style them to make them visually appealing. You can use custom checkbox styles or icon fonts for a modern look.
- **Button Styling:** Style buttons for adding, editing, and deleting tasks to make them visually prominent. Use consistent colors and sizes for buttons throughout the app to maintain coherence.

JavaScript Functionality:

The JavaScript adds dynamic behavior to the to-do list, covering task addition, completion marking, and deletion.

Adding Tasks (add function):

- Select the Form Element: Use JavaScript to select the form element where users input their tasks.
- Create Task Element: Create a new list item element (li) to represent the task.
- Set Task Content: Set the inner text or HTML of the list item to the task input value.

Marking Tasks as Completed:

- Add Event Listener: Attach a click event listener to each task element so that when a user clicks on a task, the completion status can be toggled.
- Toggle Completion Status: Define a function that toggles the completion status of a task. This function should add or remove a CSS class to visually indicate whether the task is completed or not.

Removing Tasks:

- Remove Completed Tasks: Create a function to remove completed tasks from the list. When a task is marked as completed, call this function to remove it from the displayed list. You can do this by targeting the specific task element and removing it from the DOM.

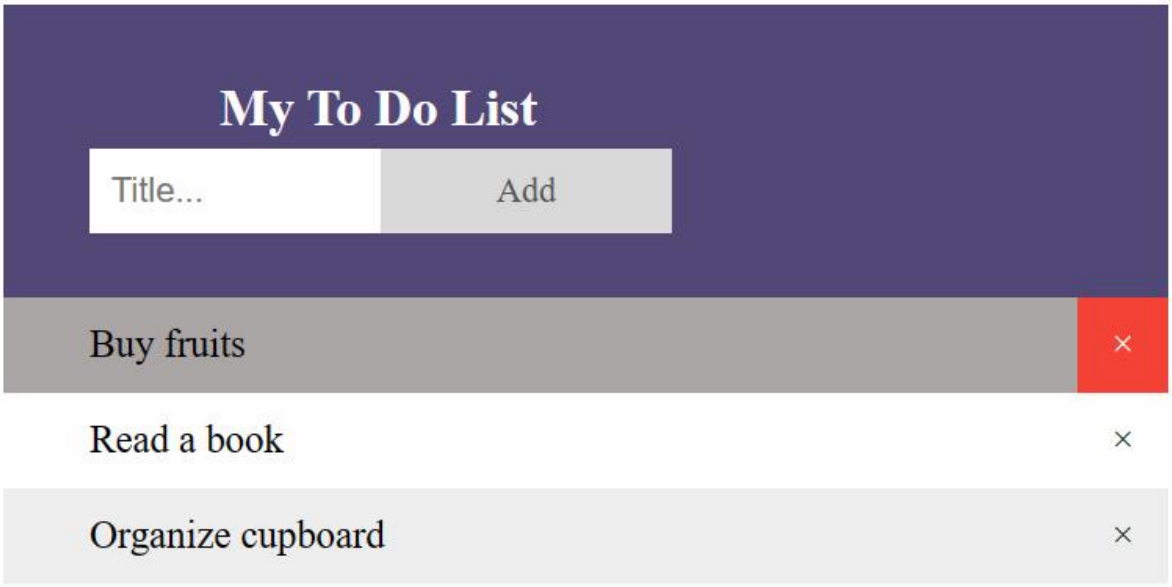
USAGE

- **Form Submission Handling:** Use JavaScript to listen for the submission of the task input form.
- **Access Task Input:** Retrieve the value entered by the user in the task input field.
- **Update Display:** Update the user interface to reflect the addition of the new task. This involves adding a new list item to the DOM with the task description and any other relevant information.

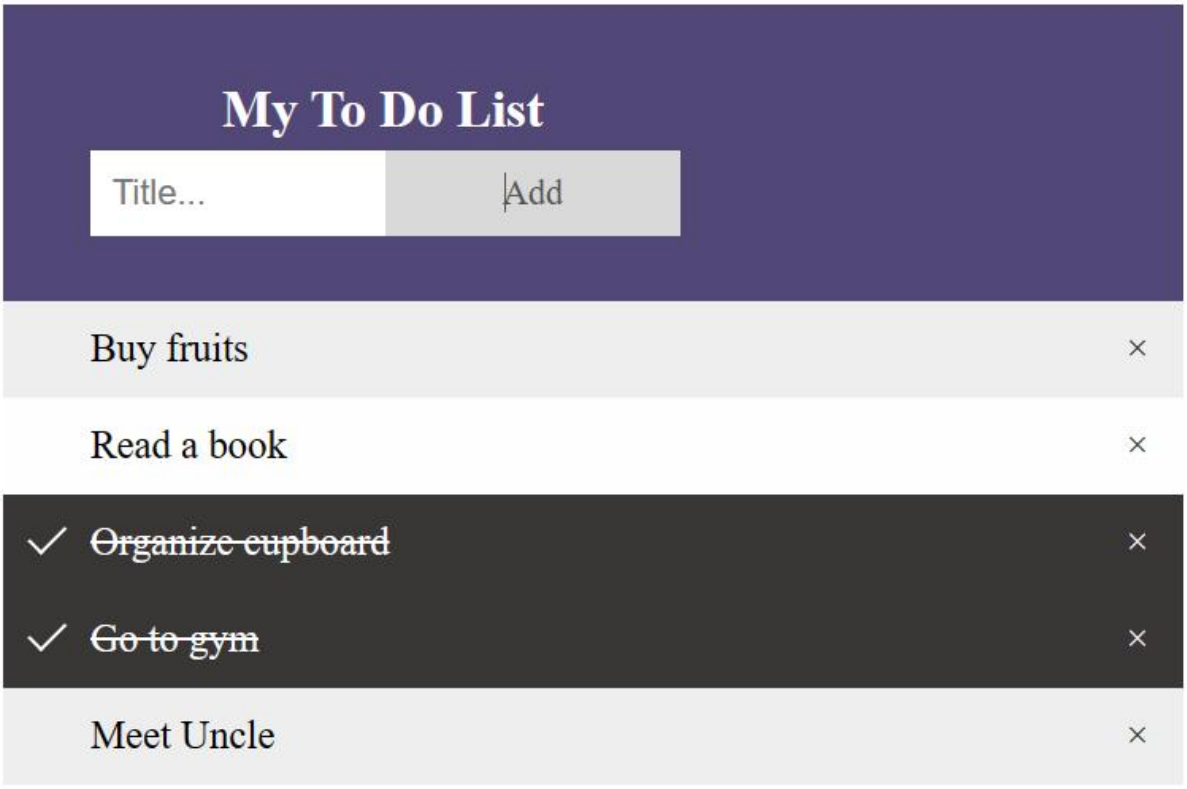
CONCLUSION

the To-Do List app serves as a valuable tool for organizing tasks and managing daily activities efficiently. By combining HTML, CSS, and JavaScript, we've created a user-friendly interface that allows users to seamlessly add, edit, and remove tasks. The app's simplicity and effectiveness lie in its intuitive design, dynamic updates, and responsive functionality. With features such as task management logic, event handling, and accessibility considerations, the app caters to diverse user needs and preferences. Whether it's keeping track of work assignments, household chores, or personal goals, the To-Do List app empowers users to stay organized and productive, ultimately enhancing their daily lives.

OUTPUT


My To Do List
Title...
Add
Buy fruits
Read a book
Organize cupboard


My To Do List
Title...
Add
Buy fruits
Read a book
Organize cupboard
Go to gym
Meet Uncle


My To Do List
Title...
Add
Buy fruits
Read a book
Organize cupboard
Go to gym
Meet Uncle