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

**INTERN INFORMATION:**

**Name:** Namrata Kiran Tamse

**ID:** C0D7067

**INTRODUCTION**

In the realm of personal productivity and organizational tools, the to-do list occupies a central role, universally recognized for its simplicity and effectiveness in managing tasks and priorities. The evolution from paper-based lists to digital platforms has significantly expanded the functionality and accessibility of to-do lists, making them an indispensable tool for individuals looking to optimize their time and manage their responsibilities efficiently. This project aims to further innovate in this space by developing a To-Do List application utilizing the power of modern web technologies: JavaScript, HTML, and CSS.

The transition to digital to-do lists has opened a plethora of possibilities for customization, real-time collaboration, and accessibility across devices. Despite the availability of numerous applications in the market, there remains a considerable opportunity to create a more intuitive, user-friendly, and flexible tool that caters to the varied needs of users. This project is driven by the vision to harness the capabilities of JavaScript and the versatility of web technologies to deliver superior task management experience.

**Implementation**

* JavaScript Framework: Utilize a modern JavaScript framework for building the frontend application.
* HTML/CSS: Use HTML5 and CSS3 for structuring and styling the user interface, ensuring compatibility with various web browsers.
* Responsive Design: Implement responsive design principles to ensure optimal viewing experience across desktop and mobile devices.
* User Interface Components: Utilize UI libraries for designing interactive and visually appealing components.

**CODE: -**

**HTML: -**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Simple To-Do App</title>

    <link rel="stylesheet" href="./style.css">

</head>

<body>

    <div class="container">

        <h1>Simple To-Do App</h1>

        <input type="text" id="taskInput" placeholder="Add a new task...">

        <button onclick="addTask()">Add Task</button>

        <ul id="taskList"></ul>

    </div>

    <script src="script.js"></script>

</body>

</html>

**CSS: -**

body {

    font-family: Arial, sans-serif;

}

.container {

    max-width: 400px;

    margin: 50px auto;

    padding: 20px;

    border: 1px solid #ccc;

    border-radius: 5px;

    background-color: #f9f9f9;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

input[type="text"] {

    width: 70%;

    padding: 10px;

    margin-right: 10px;

    border: 1px solid #ccc;

    border-radius: 3px;

    font-size: 16px;

}

button {

    padding: 10px 20px;

    border: none;

    border-radius: 3px;

    background-color: #4CAF50;

    color: white;

    font-size: 16px;

    cursor: pointer;

    margin-top: 30px;

}

ul {

    list-style-type: none;

    padding: 0;

}

li {

    margin-bottom: 10px;

    padding: 10px;

    background-color: #f2f2f2;

    border-radius: 3px;

    display: flex;

    align-items: center;

}

li:hover {

    background-color: #e5e5e5;

}

.done {

    text-decoration: line-through;

    color: #888;

}

**JS: -**

function addTask() {

    var taskInput = document.getElementById("taskInput");

    var taskList = document.getElementById("taskList");

    if (taskInput.value.trim() === "") {

        alert("Please enter a task.");

        return;

    }

    var li = document.createElement("li");

    li.innerText = taskInput.value;

    li.addEventListener("click", function() {

        this.classList.toggle("done");

    });

    taskList.appendChild(li);

    taskInput.value = "";

}

**CODE EXPLAINATION**

**HTML Structure:**

* This is a basic HTML document structure defining a simple To-Do App interface.
* The document includes meta tags for character set and viewport settings.
* It links an external CSS file named "style.css" for styling the app.
* Within the body, there's a container div with the class "container".
* Inside the container, there's an h1 element displaying the title "Simple To-Do App".
* An input field with the id "taskInput" is provided for users to add new tasks.
* A button labeled "Add Task" is included, triggering the "addTask()" function onclick.
* A <ul> element with the id "taskList" serves as the container for displaying tasks.
* An external JavaScript file named "script.js" is linked for handling app functionality.
* Overall, this HTML code sets up the structure for a basic To-Do App with input field, button, and task list.

**CSS Styling:**

* @import: Imports Google Font "Poppins" with various weights.
* Universal Selector (\*) resets margin, padding, and box-sizing.
* Sets default font family to "Poppins" with a fallback to sans-serif.
* Body Styles use CSS Grid to center content both horizontally and vertically.
* Applies a minimum height of 100vh to fill the screen.
* Sets background color for the body.
* .box Styles define container styles for the To-Do List.
* Styles .box h2 for heading properties.
* .box span sets text color to white.
* #mainBx Styles define input field properties

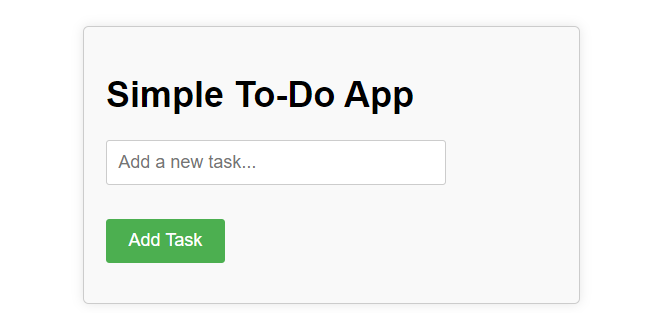
**JavaScript Functionality:**

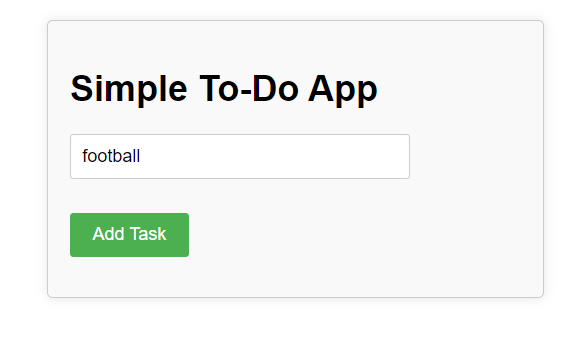
* The addTask() function is triggered when the user clicks the "Add Task" button.
* It retrieves the input field and the task list element from the HTML document.
* It checks if the input field is empty or contains only whitespace.
* If the input field is empty, it displays an alert message prompting the user to enter a task.
* Otherwise, it creates a new list item element (`li`) and sets its text content to the value entered in the input field.
* 6. It adds an event listener to the newly created list item, toggling the "done" class when clicked.
* The "done" class changes the appearance of the task item to indicate it's completed.
* The new list item is appended to the task list.
* Finally, it resets the input field to an empty state for the user to add another task.
* This function ensures tasks are added dynamically to the list with the ability to mark them as done by clicking on them.

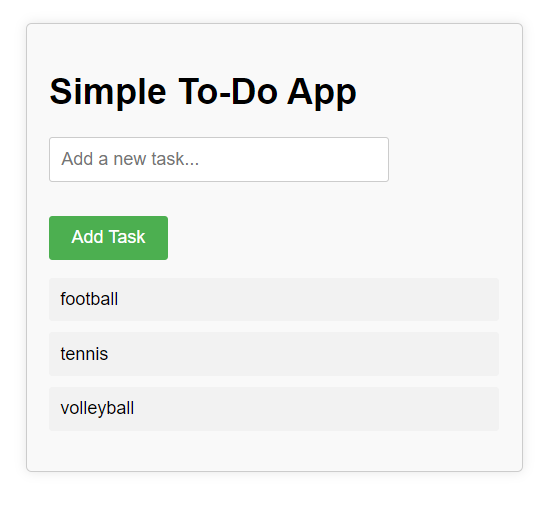
**CONCLUSION: -**

The To-Do app provides a simple and intuitive way to manage tasks effectively. With its clean interface and responsive design, users can easily add, view, and mark tasks as done. The use of Google Font "Poppins" enhances readability and visual appeal. The functionality of adding tasks dynamically and marking them as done with just a click streamlines the task management process. Overall, it's a user-friendly tool for organizing tasks and increasing productivity.

**OUTPUT: -**

****

****

****