**Assignment: Build a "To Do" Application using JavaScript**

**Objective**

This assignment aims to give you an understanding of JavaScript, HTML, and CSS by developing a simple "To Do" application. This will enhance your skills in DOM manipulation, event handling, and understanding of client-side scripting using JavaScript.

**Requirements**

* You should have a basic understanding of HTML, CSS, and JavaScript.
* Code editor such as Visual Studio Code, Sublime Text, or Atom.
* A modern web browser.

**Instructions**

**Task 1: Set Up Your Project**

1. Create a new directory for your project, name it **ToDoApp**.
2. Inside the **ToDoApp** directory, create three files: **index.html**, **style.css**, **app.js**.

**Task 2: Building HTML structure**

Open **index.html** and create a simple HTML structure. You need a header, input field for new tasks, a button for adding tasks, and an area to display the tasks.

Example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>To Do App</title>

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

</head>

<body>

<header>

<h1>To Do App</h1>

</header>

<section class="todo-container">

<input type="text" id="task-input" placeholder="New task...">

<button id="add-task-button">Add Task</button>

<ul id="todo-list">

</ul>

</section>

<script src="app.js"></script>

</body>

</html>

**Task 3: Styling your HTML**

Open **style.css** and write some styles to make your app look good.

Example:

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

padding: 20px;

}

header {

text-align: center;

}

.todo-container {

max-width: 500px;

margin: 0 auto;

}

#task-input {

width: 70%;

padding: 10px;

}

#add-task-button {

width: 25%;

padding: 10px;

background-color: #5C6BC0;

color: white;

border: none;

}

ul {

list-style: none;

padding: 0;

}

**Task 4: Adding Functionality with JavaScript**

Open **app.js** and start writing the JavaScript code to add functionality to your app.

1. First, you need to get the DOM elements you will work with.

let taskInput = document.getElementById("task-input");

let addTaskButton = document.getElementById("add-task-button");

let todoList = document.getElementById("todo-list");

1. Write a function to add a new task to the list.

function addTask() {

let taskText = taskInput.value;

if (taskText === "") {

alert("Please enter a task.");

return;

}

let newTask = document.createElement("li");

newTask.textContent = taskText;

todoList.appendChild(newTask);

taskInput.value = "";

}

addTaskButton.addEventListener("click", addTask);

**Task 5: Advanced Features**

These are some optional features you can add to your app:

* Add a "Delete" button to each task to remove it from the list.
* Upon double clicking on a task, li text should be in editable text field format. Edit and update back to list (source <https://todomvc.com/examples/react/#/> )
* ***Save the task list to localStorage so that it is not lost when the page is refreshed.***
* Add a "Complete" button to each task to mark it as done. You can add a strikethrough style to completed tasks.
* Use sample output source from ( <https://todomvc.com/examples/react/#/> )

**Evaluation Criteria**

* Correctness of HTML, CSS, and JavaScript syntax
* Functionality of the "To Do" application
* Cleanliness and organization of the code (es6 modules or common js module)
* Use of JavaScript for DOM manipulation and event handling
* The aesthetic appeal of the application (CSS styling like <https://todomvc.com/examples/react/#/> )
* Advanced features like edit / delete