```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>To-Do List Application</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 20px;
            display: flex;
            flex-direction: column;
            align-items: center;
        }
        h1 {
            color: #333;
        }
        #taskInput {
            padding: 10px;
            width: 300px;
            border: 2px solid #ccc;
            border-radius: 5px;
            margin-right: 10px;
        }
        #addTaskButton {
            padding: 10px 15px;
            background-color: #28a745;
```

```
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}
#addTaskButton:hover {
    background-color: #218838;
}
ul {
    list-style: none;
    padding: 0;
    width: 100%;
    max-width: 400px;
    margin-top: 20px;
}
li {
    background: #fff;
    padding: 10px;
    border: 1px solid #ddd;
    border-radius: 5px;
    margin: 5px 0;
    display: flex;
    align-items: center;
    justify-content: space-between;
}
button {
    background-color: #dc3545;
    color: white;
    border: none;
```

```
            border-radius: 5px;
            padding: 5px 10px;
            cursor: pointer;
        }
        button:hover {
            background-color: #c82333;
        }
        input[type="checkbox"] {
            margin-right: 10px;
        }
        .completed {
            text-decoration: line-through;
            color: #aaa;
        }
    </style>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/canvas-confetti/1.5.1/confetti.min.js"></script>
</head>
<body>
    <h1>To-Do List</h1>
    <div style="display: flex; align-items: center;">
        <input type="text" id="taskInput" placeholder="Enter your task">
        <button id="addTaskButton">Add Task</button>
    </div>
    <ul id="taskList"></ul>

    <script>
        // Function to get tasks from local storage
        const getTasks = () => {
            const tasks = localStorage.getItem('tasks');
```

```
    return tasks ? JSON.parse(tasks) : [];
};

// Function to save tasks to local storage
const saveTasks = (tasks) => {
    localStorage.setItem('tasks', JSON.stringify(tasks));
};

// Function to display tasks
const displayTasks = () => {
    const tasks = getTasks();
    const taskList = document.getElementById('taskList');
    taskList.innerHTML = '';

    tasks.forEach((task, index) => {
        const listItem = document.createElement('li');

        // Create checkbox
        const checkbox = document.createElement('input');
        checkbox.type = 'checkbox';
        checkbox.checked = task.completed;
        checkbox.onchange = () => {
            task.completed = checkbox.checked;
            saveTasks(tasks);
            displayTasks();
            // Trigger confetti if the task is completed
            if (task.completed) {
                launchConfetti();
            }
```

```
        };

        // Create task text
        const taskText = document.createElement('span');
        taskText.textContent = task.text;
        if (task.completed) {
            taskText.classList.add('completed');
        }

        // Create remove button
        const removeButton = document.createElement('button');
        removeButton.textContent = 'Remove';
        removeButton.onclick = () => {
            removeTask(index);
        };

        listItem.appendChild(checkbox);
        listItem.appendChild(taskText);
        listItem.appendChild(removeButton);
        taskList.appendChild(listItem);
    });
};

// Function to add a task
const addTask = () => {
    const taskInput = document.getElementById('taskInput');
    const taskText = taskInput.value.trim();

    if (taskText) {
```

```
        const tasks = getTasks();
        tasks.push({ text: taskText, completed: false });
        saveTasks(tasks);
        taskInput.value = '';
        displayTasks();
    } else {
        alert('Please enter a task.');
    }
};

// Function to remove a task
const removeTask = (index) => {
    const tasks = getTasks();
    tasks.splice(index, 1);
    saveTasks(tasks);
    displayTasks();
};

// Function to launch confetti
const launchConfetti = () => {
    const duration = 5 * 1000; // 5 seconds
    const animationEnd = Date.now() + duration;
    const defaults = { startVelocity: 30, spread: 360, ticks: 60, zIndex: 0 };

    function randomInRange(min, max) {
        return Math.random() * (max - min) + min;
    }

    (function frame() {
```

```
            const timeLeft = animationEnd - Date.now();
            if (timeLeft <= 0) return;

            const particleCount = 50 * (timeLeft / duration);
            confetti({
               ...defaults,
               particleCount: Math.floor(particleCount),
               origin: {
                  x: Math.random(),
                  // Since the confetti falls down, we start it from 0.2 to 0.8
                  y: Math.random() - 0.2
               }
            });

            requestAnimationFrame(frame);
         })();
      };

      // Event listener for the add task button
      document.getElementById('addTaskButton').onclick = addTask;

      // Display tasks on page load
      displayTasks();
   </script>
</body>
</html>
```