

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Task Checklist</title>
  <style>
    body {
      font-family: Tahoma, sans-serif;
      font-size: 12pt;
      line-height: 1.3;
    }
    h1 {
      font-size: 18pt;
      color: blue;
      text-align: center;
    }
    .instructions {
      width: 90%;
      margin: 0 auto;
      padding: 10px;
      background-color: white;
      text-align: justify;
    }
    .container {
      width: 90%;
      margin: 0 auto;
      border: 1px solid black;
      padding: 10px;
      background-color: white;
    }
    .progress-bar {
      width: 100%;
      background-color: #f3f3f3;
      border: 1px solid black;
    }
    .progress-bar-inner {
      height: 20px;
      width: 0%;
      background-color: royalblue;
    }
    .progress-percentage {
      text-align: center;
      padding: 10px;
    }
    table {
      width: 90%;
      margin-top: 20px;
    }
    th {
      text-align: left;
```

```

}
th, td {
  padding: 10px;
}
.task-row {
  cursor: move;
}
.buttons {
  display: flex;
  justify-content: space-between;
  margin-top: 20px;
}
.buttons button {
  font-size: 12pt;
  font-family: Tahoma, sans-serif;
  padding: 10px;
}
</style>
</head>
<body>
  <h1>Task Checklist</h1>
  <div class="instructions">Press the <b>Enter Task</b> button and type in your task. Press
  <b>OK</b> to add it to the table. Once the tasks are in the table, you can re-order them by
  dragging up and down. When you complete a task, click the checkbox and the progress bar will
  alter. Your computer will remember where you left off when you return to the task. Press
  <b>Reset</b> to clear the table and start again.</div>
  <div class="container">
    <div class="progress-bar">
      <div class="progress-bar-inner"></div>
    </div>
    <div class="progress-percentage">0%</div>
    <table>
      <thead>
        <tr>
          <th style="width: 70%;">Task to do</th>
          <th style="width: 10%;">Completed?</th>
        </tr>
      </thead>
      <tbody id="task-list">
        <!-- Tasks will be added here -->
      </tbody>
    </table>
    <div class="buttons">
      <button id="enter-task">Enter Task</button>
      <button id="reset">Reset</button>
    </div>
  </div>

  <script>
    document.addEventListener('DOMContentLoaded', function () {
      const taskList = document.getElementById('task-list');

```

```

const progressBarInner = document.querySelector('.progress-bar-inner');
const progressPercentage = document.querySelector('.progress-percentage');
const enterTaskButton = document.getElementById('enter-task');
const resetButton = document.getElementById('reset');

let tasks = JSON.parse(localStorage.getItem('tasks')) || [];
let progress = 0;

function updateProgress() {
  const completedTasks = tasks.filter(task => task.completed).length;
  const totalTasks = tasks.length;
  progress = totalTasks ? (completedTasks / totalTasks) * 100 : 0;
  progressBarInner.style.width = `${progress}%`;
  progressPercentage.textContent = `${Math.round(progress)}%`;
  if (progress === 100) {
    setTimeout(() => {
      alert('Well done, you have done all the items on the list. You may want to take a
printscreens as evidence before resetting the list');
    }, 100);
  }
}

function saveTasks() {
  localStorage.setItem('tasks', JSON.stringify(tasks));
}

function loadTasks() {
  taskList.innerHTML = "";
  tasks.forEach((task, index) => {
    const row = document.createElement('tr');
    row.className = 'task-row';
    row.draggable = true;

    const taskCell = document.createElement('td');
    taskCell.textContent = task.name;
    taskCell.style.fontSize = '14pt';

    const completedCell = document.createElement('td');
    const checkbox = document.createElement('input');
    checkbox.type = 'checkbox';
    checkbox.checked = task.completed;
    checkbox.style.width = '30px';
    checkbox.style.height = '30px';

    checkbox.addEventListener('change', () => {
      task.completed = checkbox.checked;
      saveTasks();
      updateProgress();
    });

    completedCell.appendChild(checkbox);
  });
}

```

```

    row.appendChild(taskCell);
    row.appendChild(completedCell);
    taskList.appendChild(row);

    row.addEventListener('dragstart', (e) => {
        e.dataTransfer.setData('text/plain', index);
    });

    row.addEventListener('dragover', (e) => {
        e.preventDefault();
    });

    row.addEventListener('drop', (e) => {
        const draggedIndex = e.dataTransfer.getData('text/plain');
        const targetIndex = index;
        const draggedTask = tasks[draggedIndex];
        tasks.splice(draggedIndex, 1);
        tasks.splice(targetIndex, 0, draggedTask);
        saveTasks();
        loadTasks();
    });
}

enterTaskButton.addEventListener('click', () => {
    const taskName = prompt('Enter your task:');
    if (taskName) {
        tasks.push({ name: taskName, completed: false });
        saveTasks();
        loadTasks();
        updateProgress();
    }
});

resetButton.addEventListener('click', () => {
    tasks = [];
    saveTasks();
    loadTasks();
    updateProgress();
});

window.addEventListener('beforeunload', () => {
    if (progress === 100) {
        localStorage.removeItem('tasks');
    }
});

loadTasks();
updateProgress();
});

```

```
</script>  
</body>  
</html>
```