

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>To-Do Web App</title>
  <style>
    body {
      font-family: 'Segoe UI', sans-serif;
      background: linear-gradient(to right, #e0eafc, #cfdef3);
      display: flex;
      justify-content: center;
      padding: 40px;
      animation: fadeIn 1s ease-in;
    }

    @keyframes fadeIn {
      from { opacity: 0; }
      to { opacity: 1; }
    }

    .container {
      background: #fff;
      padding: 30px;
      border-radius: 15px;
      width: 100%;
      max-width: 550px;
      box-shadow: 0 10px 30px rgba(0,0,0,0.2);
      transition: transform 0.3s ease;
    }

    .container:hover {
      transform: scale(1.02);
    }

    h1 {
      text-align: center;
      color: #2c3e50;
      margin-bottom: 20px;
      font-size: 32px;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>To-Do Web App</h1>
  </div>
</body>
</html>
```

```
}

input, button {
  padding: 12px;
  font-size: 16px;
}

.input-row {
  display: flex;
  gap: 10px;
  margin-bottom: 20px;
}

input[type="text"] {
  flex-grow: 1;
  border: 1px solid #bbb;
  border-radius: 8px;
  padding-left: 15px;
  transition: border-color 0.3s;
}

input[type="text"]:focus {
  border-color: #3498db;
  outline: none;
}

button {
  background: #27ae60;
  color: white;
  border: none;
  border-radius: 8px;
  cursor: pointer;
  transition: background 0.3s ease;
}

button:hover {
  background: #219150;
}

.task {
```

```
display: flex;
justify-content: space-between;
align-items: center;
margin: 10px 0;
padding: 12px;
border: 1px solid #ddd;
border-radius: 8px;
background: #f9f9f9;
box-shadow: 0 2px 5px rgba(0,0,0,0.05);
animation: slideUp 0.4s ease-in-out;
}

@keyframes slideUp {
  from { transform: translateY(15px); opacity: 0; }
  to { transform: translateY(0); opacity: 1; }
}

.task span {
  flex-grow: 1;
}

.task button {
  margin-left: 6px;
  padding: 8px 10px;
  border-radius: 6px;
  font-size: 14px;
}

.task button.complete {
  background: #f39c12;
}

.task button.delete {
  background: #e74c3c;
}

h2 {
  margin-top: 30px;
  color: #2c3e50;
  border-bottom: 2px solid #3498db;
```

```

padding-bottom: 5px;
}

.timestamp {
  font-size: 12px;
  color: #777;
}
</style>
</head>
<body>
  <div class="container">
    <h1> To-Do List</h1>
    <div class="input-row">
      <input type="text" id="taskInput" placeholder="Add a new task..." />
      <button onclick="addTask()">Add</button>
    </div>

    <h2>Pending Tasks</h2>
    <div id="pendingList"></div>

    <h2>Completed Tasks</h2>
    <div id="completedList"></div>
  </div>

  <script>
    let pendingTasks = [];
    let completedTasks = [];

    function addTask() {
      const input = document.getElementById('taskInput');
      const text = input.value.trim();
      if (text === '') return;
      const task = {
        text,
        created: new Date().toLocaleString()
      };
      pendingTasks.unshift(task);
      input.value = '';
      render();
    }
  </script>

```

```

function completeTask(index) {
  const task = pendingTasks.splice(index, 1)[0];
  task.completed = new Date().toLocaleString();
  completedTasks.unshift(task);
  render();
}

function deleteTask(list, index) {
  list.splice(index, 1);
  render();
}

function editTask(list, index) {
  const newText = prompt("Edit task:", list[index].text);
  if (newText !== null && newText.trim() !== '') {
    list[index].text = newText.trim();
    render();
  }
}

function render() {
  const pendingDiv = document.getElementById('pendingList');
  const completedDiv = document.getElementById('completedList');
  pendingDiv.innerHTML = '';
  completedDiv.innerHTML = '';

  pendingTasks.forEach((task, i) => {
    const div = document.createElement('div');
    div.className = 'task';
    div.innerHTML = `
      <span>
        ${task.text}<br>
        <span class="timestamp">🕒 Added: ${task.created}</span>
      </span>
      <button class="complete"
onclick="completeTask(${i})">✔</button>
      <button onclick="editTask(pendingTasks, ${i})">✎</button>
      <button class="delete" onclick="deleteTask(pendingTasks,
${i})">🗑️</button>
    `
  })
}

```

```

    `;
    pendingDiv.appendChild(div);
  });

  completedTasks.forEach((task, i) => {
    const div = document.createElement('div');
    div.className = 'task';
    div.innerHTML = `
      <span>
        ${task.text}<br>
        <span class="timestamp">🕒 Added: ${task.created} | ✅
Completed: ${task.completed}</span>
      </span>
      <button onclick="editTask(completedTasks, ${i})">✎</button>
      <button class="delete" onclick="deleteTask(completedTasks,
${i})">🗑️</button>
    `;
    completedDiv.appendChild(div);
  });
}
</script>
</body>
</html>

```

Output

