SHREYA GHOLASE
23070521145

Practical 07

To-Do List App (Add, Edit, Delete)

```
<h2>To-Do List</h2>
<input type="text" id="taskInput" placeholder="Enter a task">
<button onclick="addTask()">Add</button>

<ul id="todoList"></ul>

<script>
 function addTask() {
 const input = document.getElementById('taskInput');
const taskText = input.value.trim();
 if (!taskText) return;
 const li = document.createElement('li');
 const span = document.createElement('span');
span.textContent = taskText;

 const editBtn = document.createElement('button');
editBtn.textContent = 'Edit';
 editBtn.onclick = function () {
 const newTask = prompt('Edit task:', span.textContent);  if
(newTask !== null) span.textContent = newTask;  };

 const deleteBtn = document.createElement('button');
deleteBtn.textContent = 'Delete';
 deleteBtn.onclick = function () {
 li.remove();
 };
```

```
 li.appendChild(span);
 li.appendChild(editBtn);
 li.appendChild(deleteBtn);

 document.getElementById('todoList').appendChild(li);
input.value = '';
 }
</script>
```

Practice Tasks for To-Do List App

1. Modify the code so that when editing a task, the new task cannot be empty or just spaces.

2. Add a "Complete" button that toggles a task between *normal* and *completed* (e.g., with a strikethrough style).

3. Ensure that if a task with the same text already exists, it should not be added again.

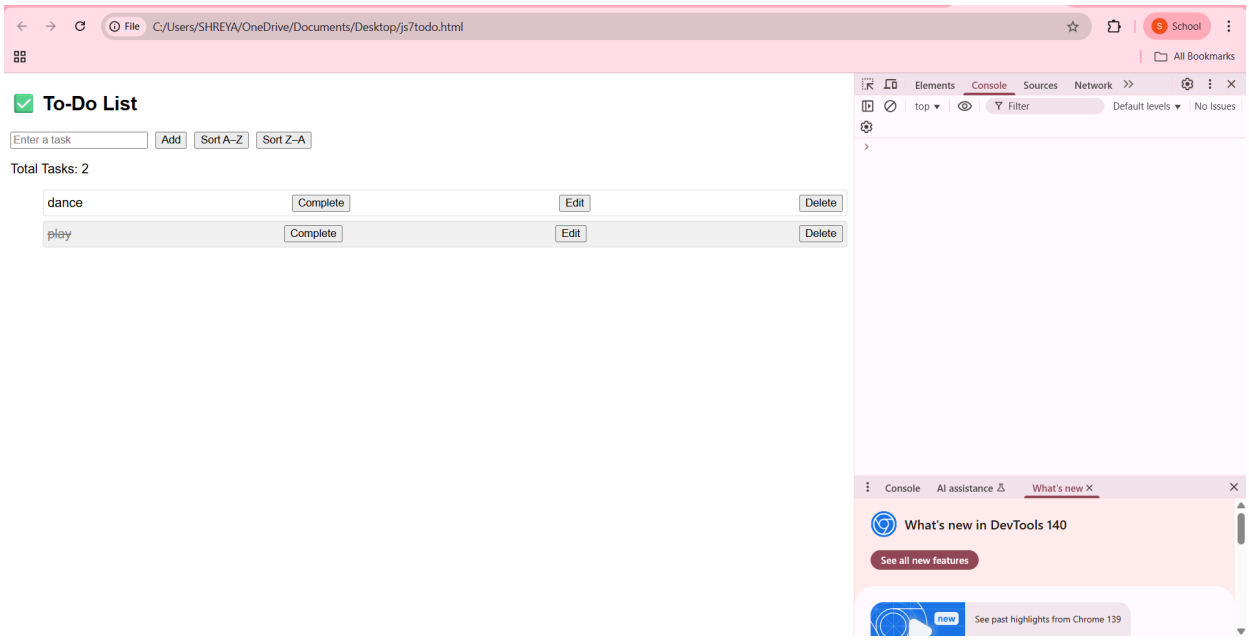
4. Display the total number of tasks and update the count dynamically when tasks are added, edited, or deleted.

5. Allow the user to press **Enter** instead of clicking the Add button to add a task. 6. Make the background color of a `<li>` change when the mouse hovers over it. 7. Show a confirmation (`confirm()`) before deleting a task to avoid accidental deletion.

8. Replace the `prompt()` with an inline `<input>` field that appears inside the list item for editing.

9. Save the tasks in `localStorage` so that they remain even after refreshing the page.

10. Add two buttons: one to sort tasks alphabetically (A–Z) and one to sort them in reverse (Z–A).




```
<!DOCTYPE html>
<html>
<head>
  <title>To-Do List App</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
    li {
      margin: 5px 0;
      padding: 5px;
      border: 1px solid #ddd;
      border-radius: 4px;
      list-style: none;
      display: flex;
      align-items: center;
      justify-content: space-between;
    }
    li span.completed {
      text-decoration: line-through;
      color: gray;
    }
```

```
    li:hover {
      background-color: #f0f0f0;
    }
    button {
      margin-left: 5px;
    }
  </style>
</head>
<body>

<h2>To-Do List</h2>
<input type="text" id="taskInput" placeholder="Enter a task">
<button onclick="addTask()">Add</button>
<button onclick="sortTasks(true)">Sort A–Z</button>
<button onclick="sortTasks(false)">Sort Z–A</button>
<p>Total Tasks: <span id="taskCount">0</span></p>
<ul id="todoList"></ul>

<script>
let tasks = JSON.parse(localStorage.getItem("tasks")) || [];
displayTasks();

// (1) Add Task
function addTask() {
  const input = document.getElementById('taskInput');
  const taskText = input.value.trim();

  if (!taskText) {
    alert("Task cannot be empty!");
    return;
  }

  // (3) Avoid duplicate tasks
  if (tasks.includes(taskText)) {
    alert("Task already exists!");
    return;
  }

  tasks.push(taskText);
```

```
  saveAndRender();
  input.value = '';
}

// Display Tasks
function displayTasks() {
  const list = document.getElementById('todoList');
  list.innerHTML = '';
  tasks.forEach((task, index) => {
    const li = document.createElement('li');

    const span = document.createElement('span');
    span.textContent = task;

    // (2) Complete button
    const completeBtn = document.createElement('button');
    completeBtn.textContent = 'Complete';
    completeBtn.onclick = function() {
      span.classList.toggle("completed");
    };

    // (8) Inline edit button
    const editBtn = document.createElement('button');
    editBtn.textContent = 'Edit';
    editBtn.onclick = function() {
      const inputEdit = document.createElement('input');
      inputEdit.type = 'text';
      inputEdit.value = span.textContent;

      inputEdit.onblur = function() {
        const newText = inputEdit.value.trim();
        if (!newText) {
          alert("Task cannot be empty!");
        } else if (tasks.includes(newText) && newText !== task) {
          alert("Task already exists!");
        } else {
          tasks[index] = newText;
          saveAndRender();
        }
```

```
      };

      li.replaceChild(inputEdit, span);
      inputEdit.focus();
    };

    // (7) Delete button with confirmation
    const deleteBtn = document.createElement('button');
    deleteBtn.textContent = 'Delete';
    deleteBtn.onclick = function() {
      if (confirm("Are you sure you want to delete this task?")) {
        tasks.splice(index, 1);
        saveAndRender();
      }
    };

    // Append all buttons and text
    li.appendChild(span);
    li.appendChild(completeBtn);
    li.appendChild(editBtn);
    li.appendChild(deleteBtn);
    list.appendChild(li);
  });

  updateCount();
}

// (4) Update task count
function updateCount() {
  document.getElementById('taskCount').textContent = tasks.length;
}

//  (9) Save to localStorage
function saveAndRender() {
  localStorage.setItem("tasks", JSON.stringify(tasks));
  displayTasks();
}

//  (10) Sort tasks
```

```
function sortTasks(ascending = true) {
  tasks.sort((a, b) => a.localeCompare(b));
  if (!ascending) tasks.reverse();
  saveAndRender();
}

//  (5) Add with Enter key
document.getElementById('taskInput').addEventListener('keypress', function(e) {
  if (e.key === 'Enter') addTask();
});

// Load tasks from storage on page open
window.onload = displayTasks;
</script>

</body>
</html>
```