

# Practical 11 - WT

## JavaScript Basics and DOM Manipulation

**Task:** Interactive To-Do List

**Objective:** Create a to-do list application with JavaScript.

**Activities:**

- Create an HTML form to add new tasks.
- Use JavaScript to add tasks to the list dynamically.
- Implement functionality to mark tasks as complete and delete tasks.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0"> <title>Interactive To-Do List</title>
<style>

  body {
    font-family: Arial, sans-serif;
  }
  .container {
    width: 300px;
    margin: 0 auto;
    padding-top: 50px;
  }
  input[type="text"] {
    width: calc(100% - 22px);
    padding: 10px;
    margin-right: 10px;
  }
  button {
    padding: 10px;
  }
  ul {
    list-style-type: none;
    padding: 0;
  }
  li {
    display: flex;
    justify-content: space-between;
    padding: 10px;
```

```
border-bottom: 1px solid #ddd;
```

```
}
```

Web Technologies 23IT020

```
.completed {
```

```
text-decoration: line-through;
```

```
color: gray;
```

```
}
```

```
.delete-btn {
```

```
background: red;
```

```
color: white;
```

```
border: none;
```

```
padding: 5px;
```

```
cursor: pointer;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<h1>To-Do List</h1>
```

```
<form id="task-form">
```

```
<input type="text" id="task-input" placeholder="Enter a new  
task"> <button type="submit">Add Task</button>
```

```
</form>
```

```
<ul id="task-list"></ul>
```

```
</div>
```

```
<script src="pr11.js"></script>
```

```
</body>
```

```
</html>
```

```
document.addEventListener('DOMContentLoaded', () => {
```

```
const form = document.getElementById('task-form');
```

```
const taskInput = document.getElementById('task-input');
```

```
const taskList = document.getElementById('task-list');
```

```
form.addEventListener('submit', (e) => {
```

```
e.preventDefault();
```

```
const taskText = taskInput.value.trim();
```

```
if (taskText === '') return;
```

```
const li = document.createElement('li');
```

```
li.textContent = taskText;
```

```
// Create the complete button
```

```
const completeButton = document.createElement('button');
completeButton.textContent = 'Complete';
completeButton.addEventListener('click', () => {
    li.classList.toggle('completed');

```

Web Technologies 23IT020

```
});
const deleteButton = document.createElement('button');
deleteButton.textContent = 'Delete';
deleteButton.classList.add('delete-btn');
deleteButton.addEventListener('click', () => {
    taskList.removeChild(li);
});

```

```
li.appendChild(completeButton);
li.appendChild(deleteButton);

```

```
taskList.appendChild(li);

```

```
taskInput.value = '';

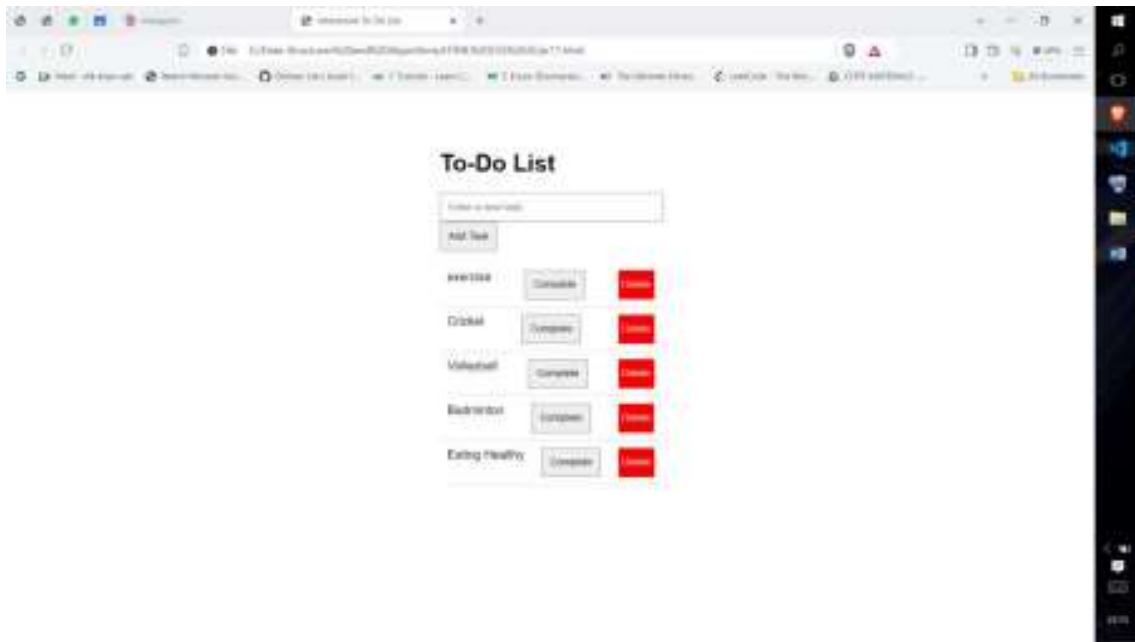
```

```
});

```

```
});

```



Learning Outcomes:- · **Understanding DOM Manipulation:**

· **Improved JavaScript Skills:**

· **Event Handling**

- **HTML Structure and Semantic Markup**
- **Styling with CSS**