```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>To do list - website</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/6.5.1/css/all.min
.css" integrity="sha512-
DTOQO9RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQeAIftl+Vegovlnee1c9QX4TctnWMn13TZy
e+gi
Mm8e2LwA==" crossorigin="anonymous" referrerpolicy="no-referrer" />
    <style>
        * {
            padding: 0;
            margin: 0;
            box-sizing: border-box;
        }

        body {
            height: 100vh;
            background: linear-gradient(#35ff50, #8b39f6, #21bde4);
        }

        .app {
            font-family: "poppins", sans-serif;
            width: min(95vw, 500px);
            position: absolute;
            margin: auto;
            left: 0;
            right: 0;
            top: 1.875em;
        }

        .container {
            padding: 2em 2.5em;
            background-color: #f1f8fb;
            box-shadow: 0 1em 2em rgba(0, 0, 0, 0.3);
```

```
        border-radius: 0.8em;
    }

    #wrapper {
        position: relative;
        display: grid;
        grid-template-columns: 8fr 4fr;
        gap: 1em;
    }

    #wrapper input {
        width: 100%;
        background-color: transparent;
        color: #111111;
        font-size: 0.9em;
        border: none;
        border-bottom: 2px solid #d1d3d4;
        padding: 1em 0.5em;
    }

    #wrapper input:focus {
        outline: none;
        border-color: #5a95ff;
    }

    #wrapper button {
        position: relative;
        border-radius: 0.5em;
        font-weight: 500;
        font-size: 1em;
        background-color: #5a95ff;
        border: none;
        outline: none;
        color: #ffffff;
        cursor: pointer;
    }

    #tasks {
        margin-top: 1em;
        border-radius: 0.5em;
```

```
        width: 100%;
        position: relative;
        padding: 1em 0.5em;
    }

    .task {
        background-color: #ffffff;
        padding: 0.8em 1em;
        display: grid;
        grid-template-columns: 1fr 8fr 2fr 2fr;
        gap: 1em;
        border-radius: 0.5em;
        box-shadow: 0 0.5em 1em rgba(0, 0, 0, 0.05);
        align-items: center;
        cursor: pointer;
    }

    .task:not(:first-child) {
        margin-top: 1em;
    }

    .task input[type="checkbox"] {
        position: relative;
        appearance: none;
        height: 20px;
        width: 20px;
        border-radius: 50%;
        border: 2px solid #e1e1e1;
    }

    .task input[type="checkbox"]:before {
        content: "";
        position: absolute;
        transform: translate(-50%, -50%);
        top: 50%;
        left: 50%;
    }

    .task input[type="checkbox"]:checked {
        background-color: #5a95ff;
```

```
        border-color: #5a95ff;
    }

    .task input[type="checkbox"]:checked::before {
        position: absolute;
        content: "\f00c";
        color: #ffffff;
        font-size: 0.8em;
        font-family: "font Awesome 5 Free";
        font-weight: 900;
    }

    .task span {
        font: 0.9em;
        font-weight: 400;
        word-break: break-all;
    }

    .task button {
        color: #ffffff;
        width: 100%;
        padding: 1em 0;
        border-radius: 0.5em;
        border: pointer;
        outline: none;
    }

    .edit {
        background-color: #5a95ff;
    }

    .delete {
        background-color: #ff5c5c;
    }

    #pending-tasks span {
        color: #5a95ff;
    }

    .completed {
```

```
            text-decoration: line-through;
            color: #a0a0a0;
        }

        #error {
            text-align: center;
            background-color: #ffffff;
            color: #ff5c5c;
            margin-top: 1.5em;
            padding: 1em 0;
            border-radius: 0.8em;
            display: none;
        }
    </style>

</head>

<body>
    <div class="app">
        <div class="container">
            <div id="wrapper">
                <input type="text" name="text" placeholder="Task to be
done.." />
                <button id="add-btn">Add</button>
            </div>
            <div id="tasks">
                <p id="pending-task">
                    You have <span class="count-value">0</span> task
                    (s) to complete.
                </p>
            </div>
        </div>
        <p id="error">Input can not be empty!</p>
    </div>
    <script>
        const addBtn = document.querySelector
            ("#add-btn");
        const newTaskInput = document.querySelector
            ("#wrapper input");
        const tasksContainer = document.querySelector
```

```
        ("#tasks");
    const error = document.getElementById
        ("error");
    const countValue = document.querySelector
        (".count-value");
    let taskCount = 0;
    const displayCount = (taskcount) => {
        countValue.innerText = taskCount;
    };
    const addTask = () => {
        const taskName = newTaskInput.value.trim();
        error.style.display = "none";
        if (!taskName) {
            setTimeout(() => {
                error.style.display = "block";
            }, 200);
            return;
        }
        const task = `<div class="task">
                    <input type = "checkbox" class="task-check">
                    <span class="taskname">${taskName}</span>
                    <button class="edit">
                    <i class="fa-solid fa-pen"></i>
                    </button>
                    <button class="delete">
                    <i class="fa-solid fa-trash"></i>
                    </button>
                    </div>`;
        tasksContainer.insertAdjacentHTML
            ("beforeend", task);
        const deleteButtons = document.querySelectorAll
            (".delete");
        deleteButtons.forEach(button => {
            button.onclick = () => {
                button.parentNode.remove();
                taskCount -= 1;
                displayCount(taskCount);
            };
        });
        const editButtons = document.querySelectorAll(".edit");
```

```
            editButtons.forEach((editBtn) => {
                editBtn.onclick = (e) => {
                    let targetElement = e.target;
                    if (!(e.target.className == "edit")) {
                        targetElement = e.target.parentElement;
                    }
                    newTaskInput.value = targetElement.
                        previousElementSibling?.innerText;
                    targetElement.parentNode.remove();
                    taskCount -= 1;
                    displayCount(taskCount);
                };
            });
            const tasksCheck = document.querySelectorAll
                (".task-check");
            tasksCheck.forEach((checkBox) => {
                checkBox.onchange = () => {

checkBox.nextElementSibling.classList.toggle("completed");
                    if (checkBox.checked) {
                        taskCount -= 1;
                    } else {
                        taskCount += 1;
                    }
                    displayCount(taskCount);
                };
            });
            taskCount += 1;
            displayCount(taskCount);
            newTaskInput.value = "";
        };
        addBtn.addEventListener("click", addTask);
        window.onload = () => {
            taskCount = 0;
            displayCount(taskCount);
            newTaskInput.value = "";
        };
    </script>
</body>
```

```
</html>
```