

Roll no: 202501100500035

Name: Animesh Raj (CSIT)

HTML CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>To-Do List App</title>


<style>

  :root {

    --bg: #0f172a;

    --card: #1e293b;

    --accent: #38bdf8;

    --text: #e2e8f0;

    --muted: #94a3b8;

    --radius: 10px;

    font-family: "Poppins", sans-serif;

  }


  body {

    margin: 0;

    background: var(--bg);

    color: var(--text);

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

    padding: 20px;

  }


  .todo-container {

    width: 380px;

    background: var(--card);

    padding: 20px;

    border-radius: var(--radius);

    box-shadow: 0 0 20px rgba(0, 0, 0, 0.4);

  }
```

```
h2 {  
  margin: 0 0 20px;  
  text-align: center;  
  color: var(--accent);  
}
```

```
.input-box {  
  display: flex;  
  gap: 8px;  
}
```

```
input {  
  flex: 1;  
  padding: 12px;  
  border-radius: var(--radius);  
  border: none;  
  outline: none;  
  font-size: 16px;  
}
```

```
button {  
  padding: 12px 18px;  
  border: none;  
  border-radius: var(--radius);  
  background: var(--accent);  
  color: black;  
  font-weight: 700;  
  cursor: pointer;  
}
```

```
ul {  
  margin-top: 20px;  
  padding: 0;  
  list-style: none;  
}
```

```
li {  
  background: #0f1724;  
  padding: 12px;  
  border-radius: var(--radius);  
  margin-bottom: 10px;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;
```

```
      transition: 0.3s;
    }

    li.completed {
      text-decoration: line-through;
      color: var(--muted);
      opacity: 0.6;
    }

    .delete-btn {
      background: #ef4444;
      padding: 6px 10px;
      border-radius: 6px;
      color: #fff;
      cursor: pointer;
      font-size: 12px;
    }
  </style>
</head>

<body>

<div class="todo-container">

  <h2>To-Do List</h2>

  <div class="input-box">

    <input type="text" id="taskInput" placeholder="Add a new task..." />

    <button onclick="addTask()">Add</button>

  </div>

  <ul id="taskList"></ul>
</div>

<script>

  const taskInput = document.getElementById("taskInput");
  const taskList = document.getElementById("taskList");

  // Load tasks from localStorage
  window.onload = () => {
    const saved = JSON.parse(localStorage.getItem("tasks")) || [];
    saved.forEach(t => renderTask(t.text, t.completed));
  };

  // Add Task
  function addTask() {
```

```

const text = taskInput.value.trim();

if (text === "") {
    alert("Please enter a task!");
    return;
}

renderTask(text, false);
saveTasks();

taskInput.value = "";
}

// Render Task
function renderTask(text, completed) {
    const li = document.createElement("li");
    li.textContent = text;

    if (completed) li.classList.add("completed");

    // Click to mark completed
    li.addEventListener("click", () => {
        li.classList.toggle("completed");
        saveTasks();
    });

    // Delete button
    const delBtn = document.createElement("span");
    delBtn.textContent = "Delete";
    delBtn.className = "delete-btn";

    delBtn.addEventListener("click", (e) => {
        e.stopPropagation();
        li.remove();
        saveTasks();
    });

    li.appendChild(delBtn);
    taskList.appendChild(li);
}

// Save tasks to localStorage
function saveTasks() {
    const tasks = [];

    document.querySelectorAll("li").forEach(li => {
        tasks.push({

```

```
        text: li.childNodes[0].textContent,
        completed: li.classList.contains("completed")
    });
});

localStorage.setItem("tasks", JSON.stringify(tasks));
}

// Enter key adds task
taskInput.addEventListener("keypress", (e) => {
    if (e.key === "Enter") addTask();
});
</script>

</body>
</html>
```

OUTPUT:

