Project Title

To-Do List App with Local Storage

Objective

- To design and develop a dynamic To-Do
- List web application that allows users to:
- Create new tasks
- Update existing tasks
- Delete tasks
- Store tasks using LocalStorage so data remains persistent even after page refresh

Tech Stack

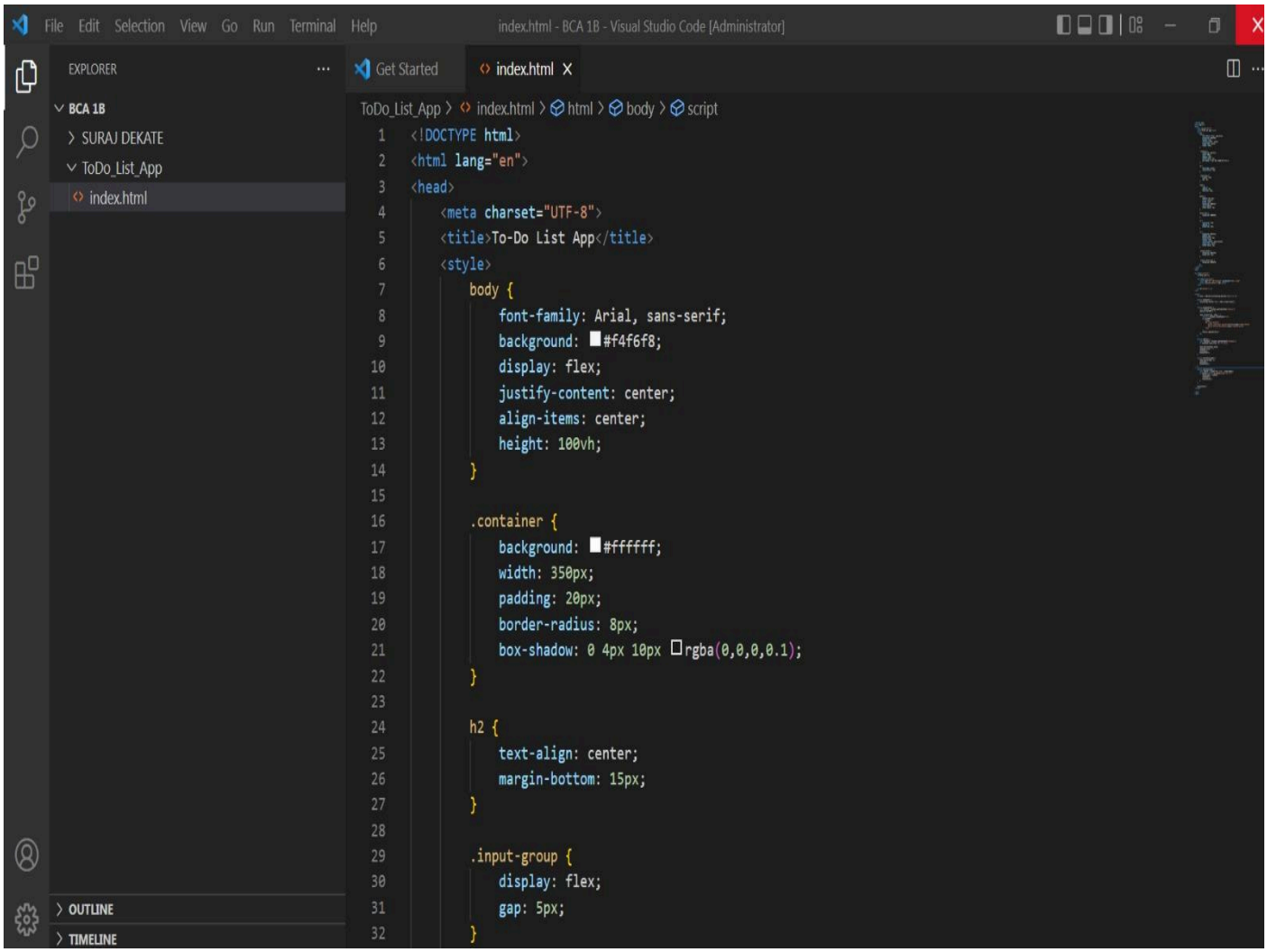
- HTML – Structure
- CSS – Styling (clean & professional UI)
- JavaScript – Logic & interactivity
- LocalStorage – Persistent data storage

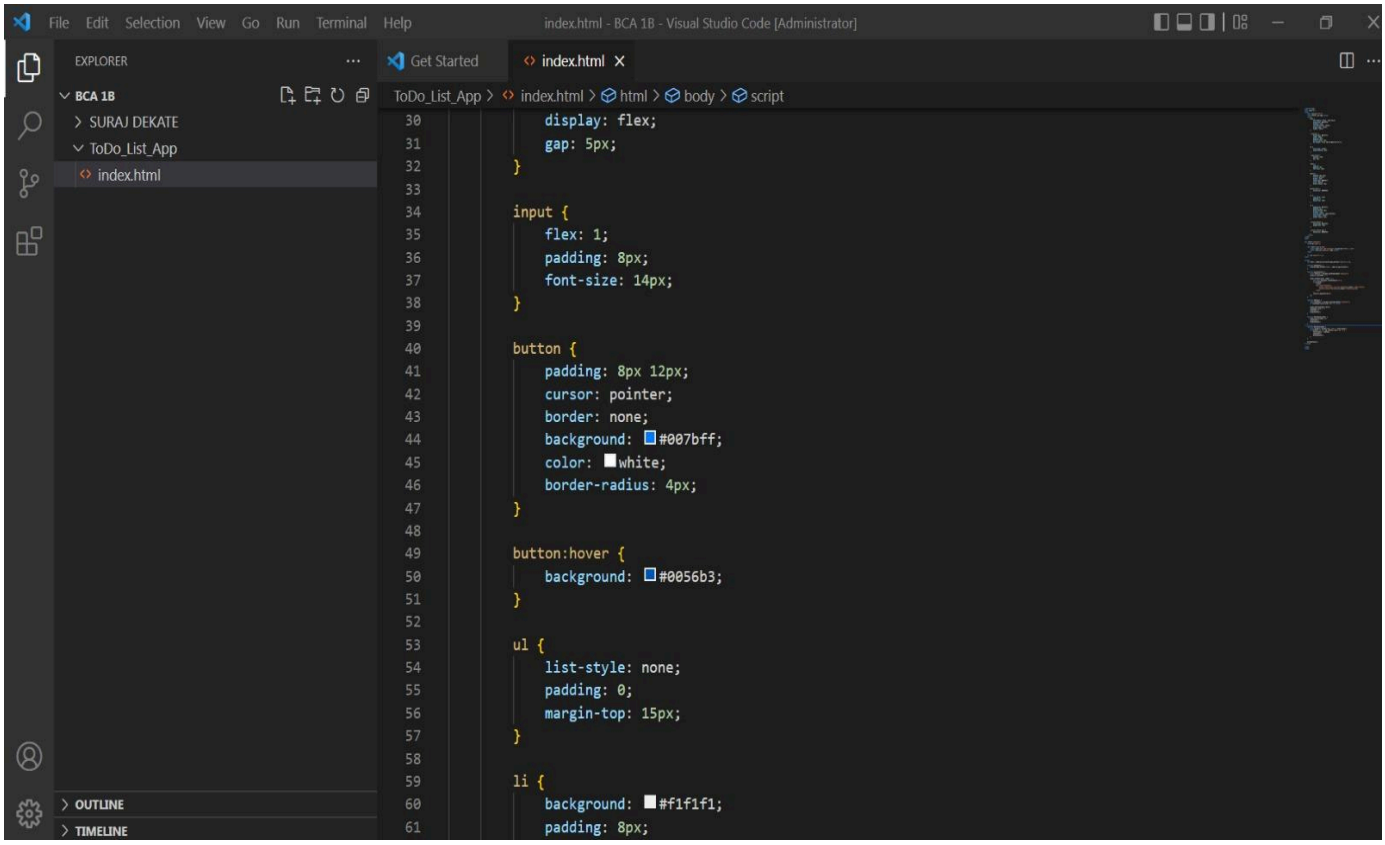
Expected Output:

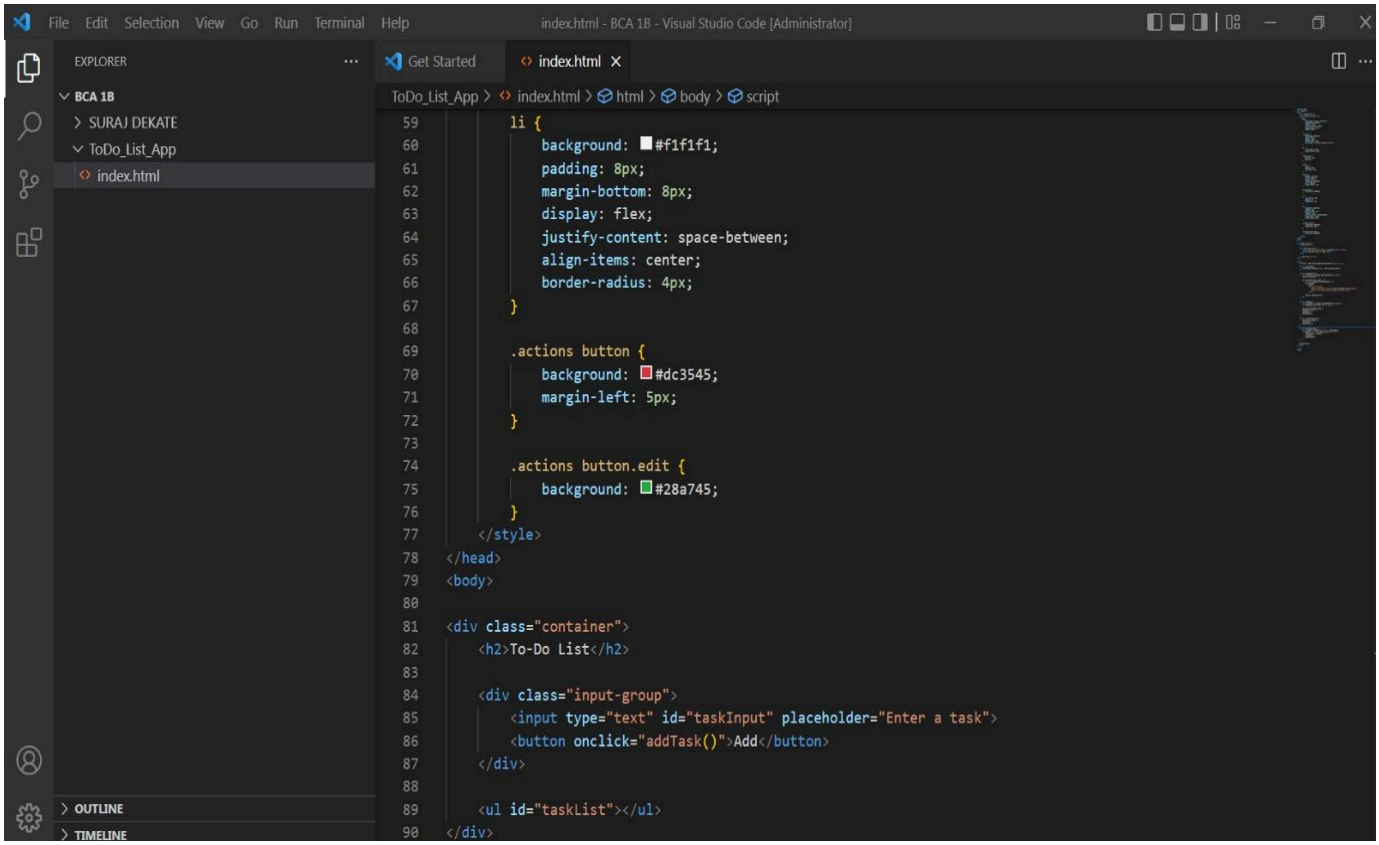
➔ User can add, edit, and delete tasks

➔ Tasks remain saved even after browser refresh

Program:




File Edit Selection View Go Run Terminal Help
index.html - BCA 1B - Visual Studio Code [Administrator]
EXPLORER
BCA 1B
SURAJ DEKATE
ToDo_List_App
index.html
Get Started
index.html
ToDo_List_App > index.html > html > body > script
display: flex;
gap: 5px;
}
input {
flex: 1;
padding: 8px;
font-size: 14px;
}
button {
padding: 8px 12px;
cursor: pointer;
border: none;
background: #007bff;
color: white;
border-radius: 4px;
}
button:hover {
background: #0056b3;
}
ul {
list-style: none;
padding: 0;
margin-top: 15px;
}
li {
background: #f1f1f1;
padding: 8px;
OUTLINE
TIMELINE

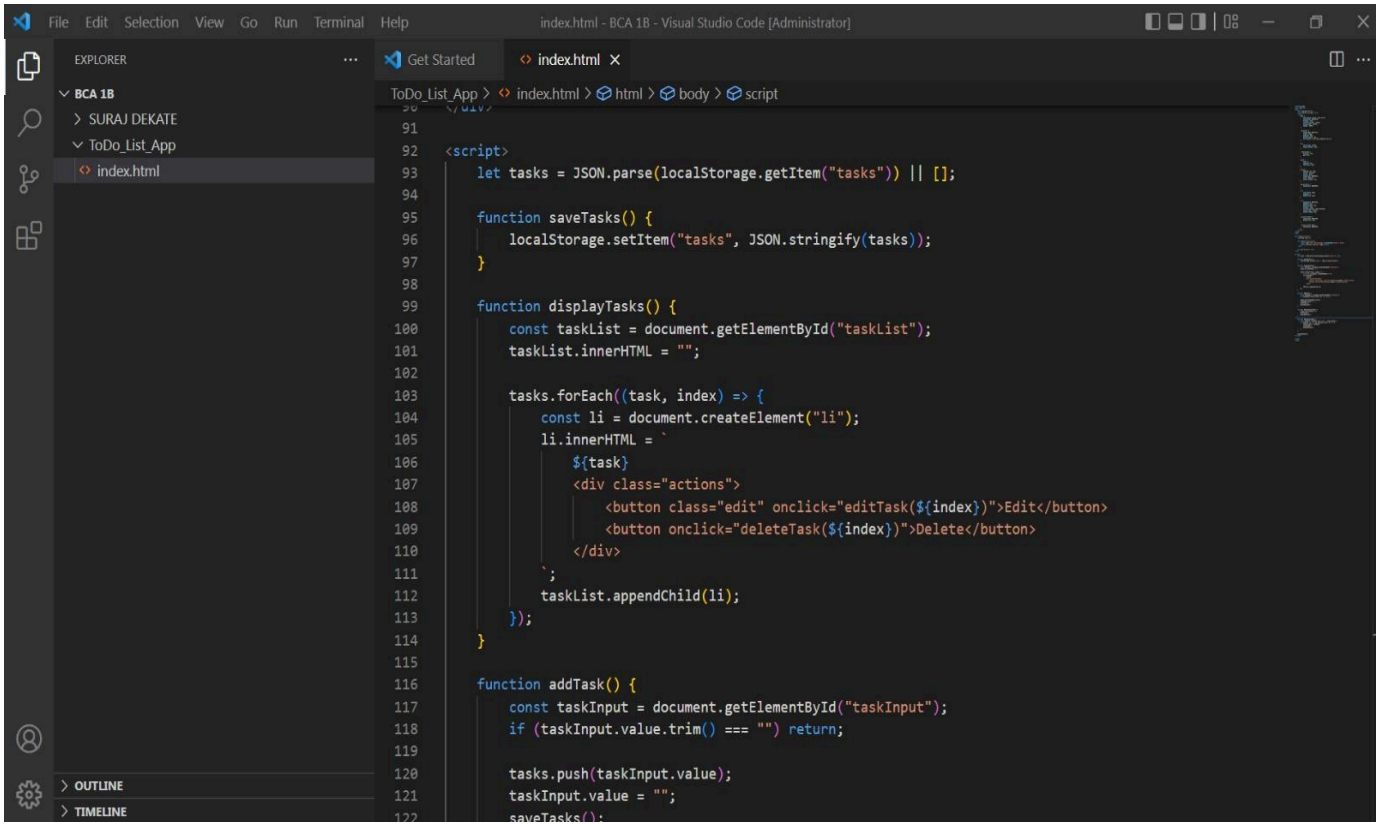

File Edit Selection View Go Run Terminal Help
index.html - BCA 1B - Visual Studio Code [Administrator]
EXPLORER
BCA 1B
SURAJ DEKATE
ToDo_List_App
index.html
Get Started
index.html
ToDo_List_App > index.html > html > body > script
li {
background: #f1f1f1;
padding: 8px;
margin-bottom: 8px;
display: flex;
justify-content: space-between;
align-items: center;
border-radius: 4px;
}
.actions button {
background: #dc3545;
margin-left: 5px;
}
.actions button.edit {
background: #28a745;
}
</style>
</head>
<body>

<h2>To-Do List</h2>

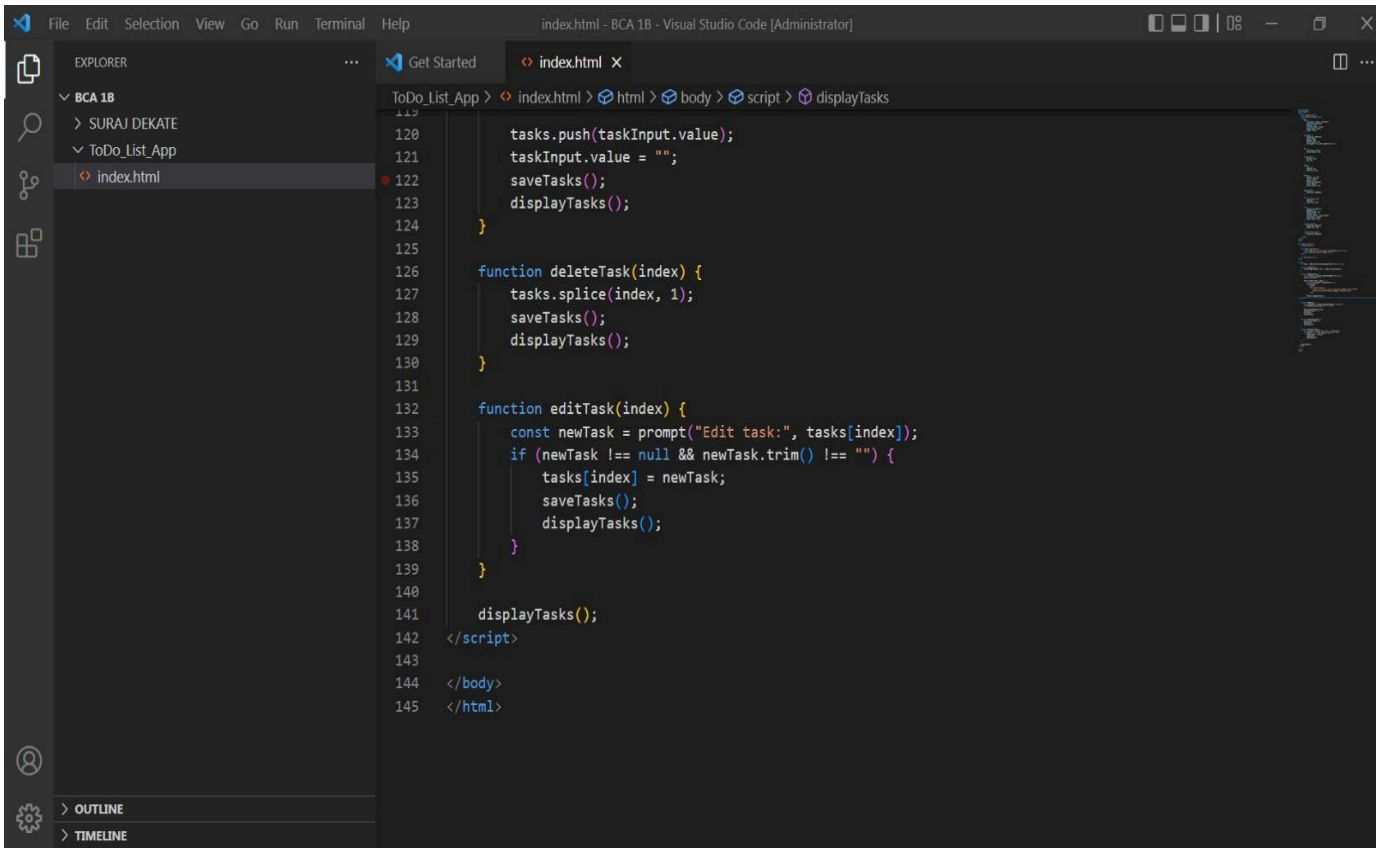
<input type="text" id="taskInput" placeholder="Enter a task">
<button onclick="addTask()">Add</button>

<ul id="taskList">

OUTLINE
TIMELINE

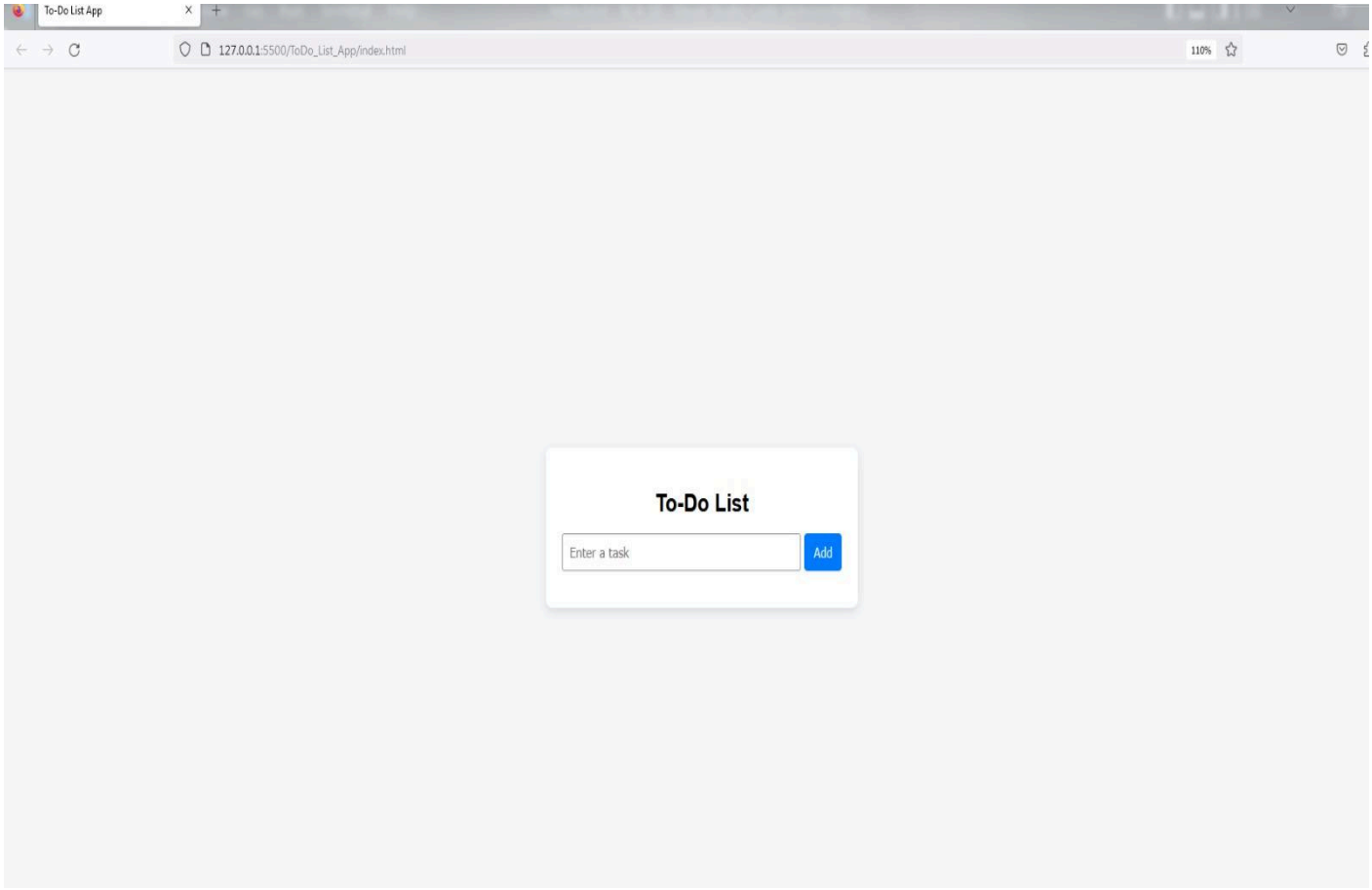



```
91
92  <script>
93      let tasks = JSON.parse(localStorage.getItem("tasks")) || [];
94
95      function saveTasks() {
96          localStorage.setItem("tasks", JSON.stringify(tasks));
97      }
98
99      function displayTasks() {
100         const taskList = document.getElementById("taskList");
101         taskList.innerHTML = "";
102
103         tasks.forEach((task, index) => {
104             const li = document.createElement("li");
105             li.innerHTML = `
106                 ${task}
107                 <div class="actions">
108                     <button class="edit" onclick="editTask(${index})">Edit</button>
109                     <button onclick="deleteTask(${index})">Delete</button>
110                 </div>
111             `;
112             taskList.appendChild(li);
113         });
114     }
115
116     function addTask() {
117         const taskInput = document.getElementById("taskInput");
118         if (taskInput.value.trim() === "") return;
119
120         tasks.push(taskInput.value);
121         taskInput.value = "";
122         saveTasks();
```




```
120         tasks.push(taskInput.value);
121         taskInput.value = "";
122         saveTasks();
123         displayTasks();
124     }
125
126     function deleteTask(index) {
127         tasks.splice(index, 1);
128         saveTasks();
129         displayTasks();
130     }
131
132     function editTask(index) {
133         const newTask = prompt("Edit task:", tasks[index]);
134         if (newTask !== null && newTask.trim() !== "") {
135             tasks[index] = newTask;
136             saveTasks();
137             displayTasks();
138         }
139     }
140
141     displayTasks();
142 </script>
143
144 </body>
145 </html>
```

Output:


To-Do List App
127.0.0.1:5500/ToDo_List_App/index.html
110%
To-Do List
Enter a task
Add