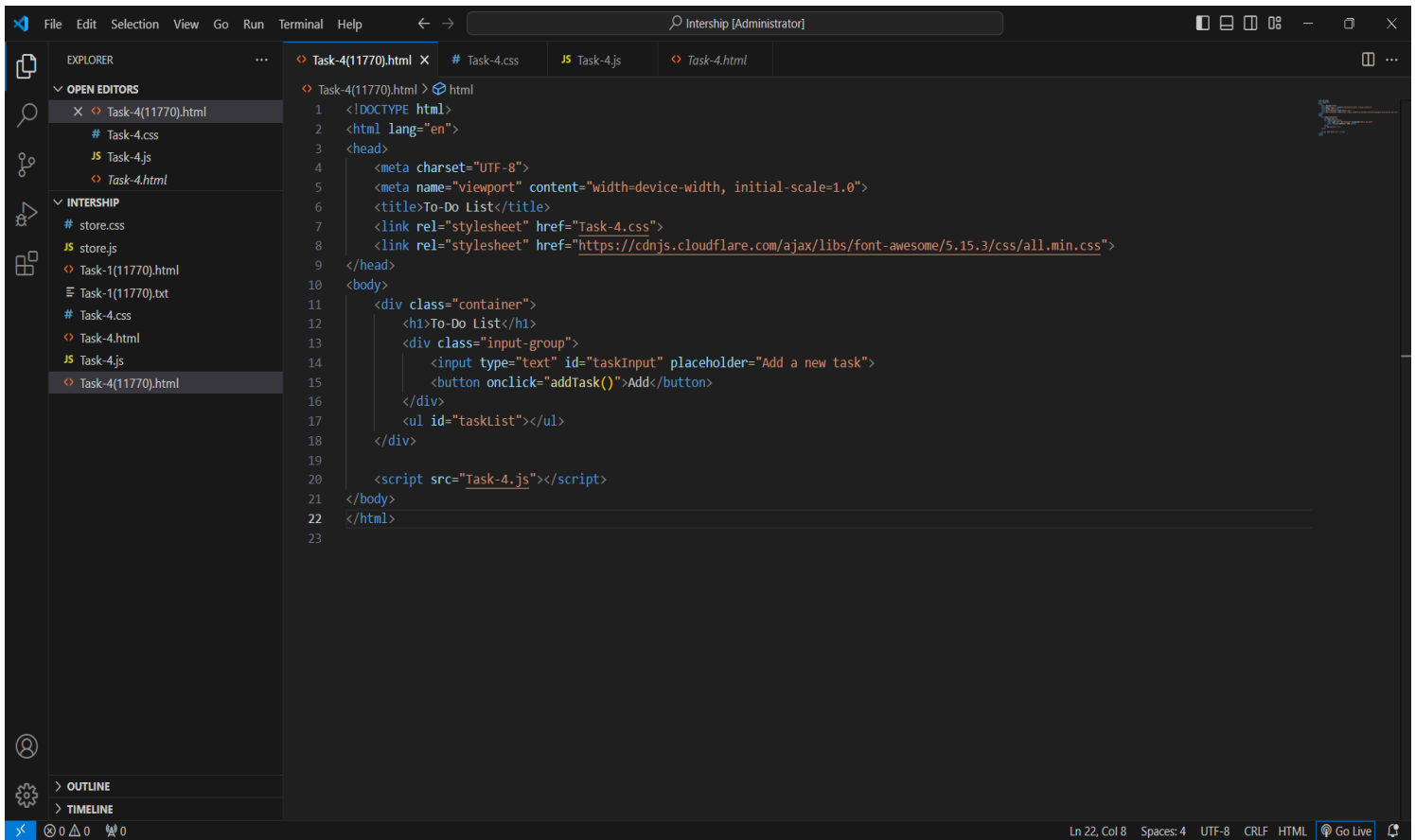


Task-4

.html

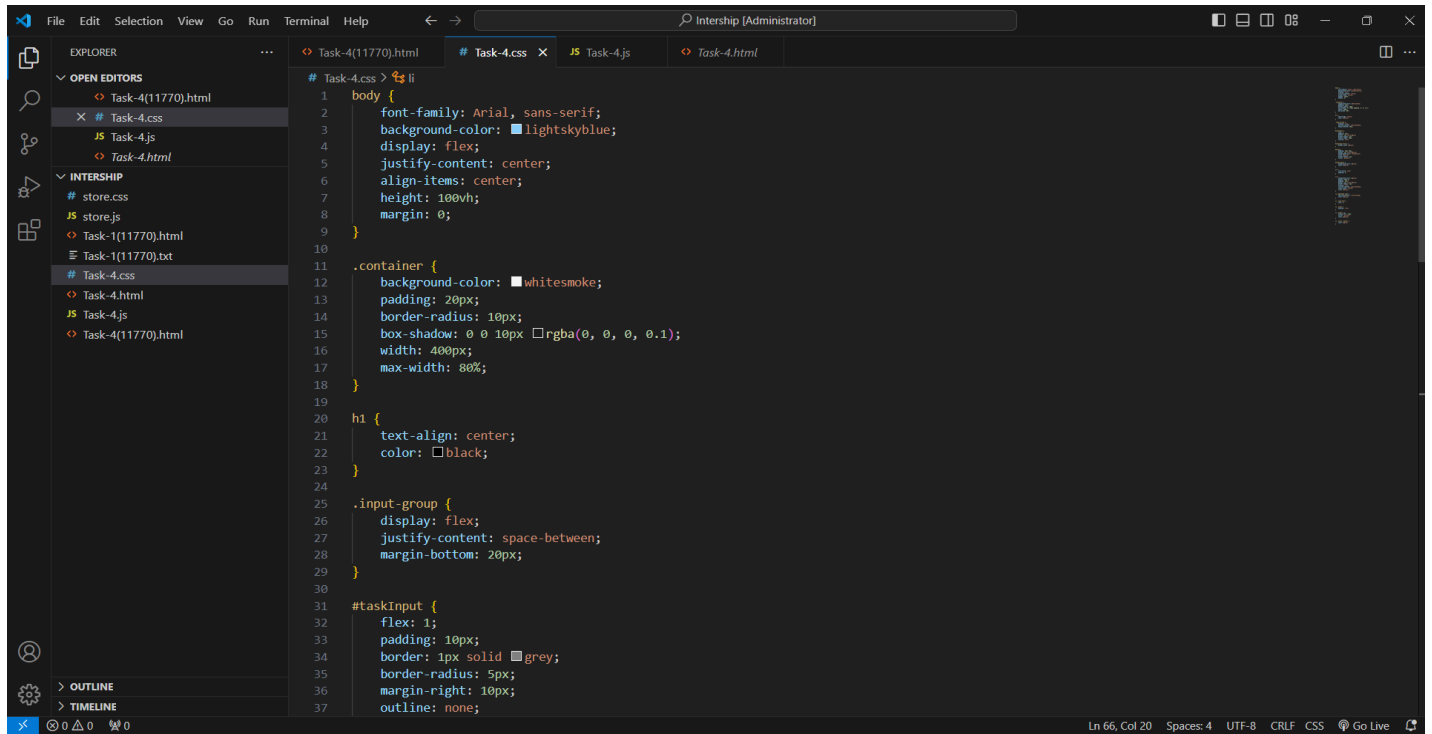


The screenshot shows a VS Code editor window with the following components:


- Explorer Panel:** Displays the file structure. The 'OPEN EDITORS' section shows 'Task-4(11770).html', 'Task-4.css', 'Task-4.js', and 'Task-4.html'. The 'INTERSHIP' section shows 'store.css', 'store.js', 'Task-1(11770).html', 'Task-1(11770).txt', 'Task-4.css', 'Task-4.html', 'Task-4.js', and 'Task-4(11770).html'.
- Editor Panel:** Shows the content of 'Task-4(11770).html'. The code is as follows:

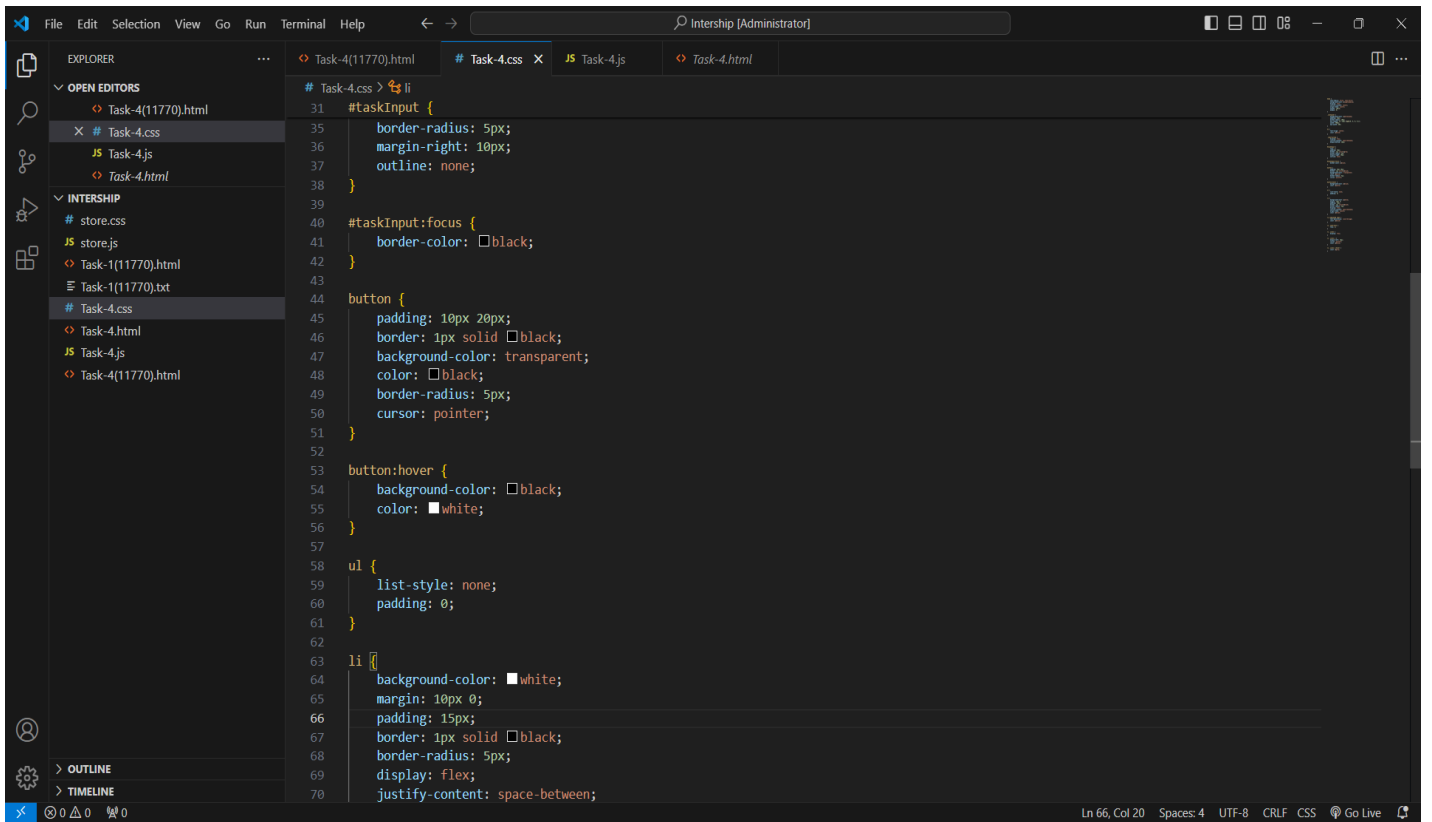
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>To-Do List</title>
7   <link rel="stylesheet" href="Task-4.css">
8   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">
9 </head>
10 <body>
11   <div class="container">
12     <h1>To-Do List</h1>
13     <div class="input-group">
14       <input type="text" id="taskInput" placeholder="Add a new task">
15       <button onclick="addTask()">Add</button>
16     </div>
17     <ul id="taskList"></ul>
18   </div>
19
20   <script src="Task-4.js"></script>
21 </body>
22 </html>
23
```
- Terminal Panel:** Empty.
- Status Bar:** Shows 'Ln 22, Col 8', 'Spaces: 4', 'UTF-8', 'CRLF', 'HTML', and 'Go Live'.

.CSS



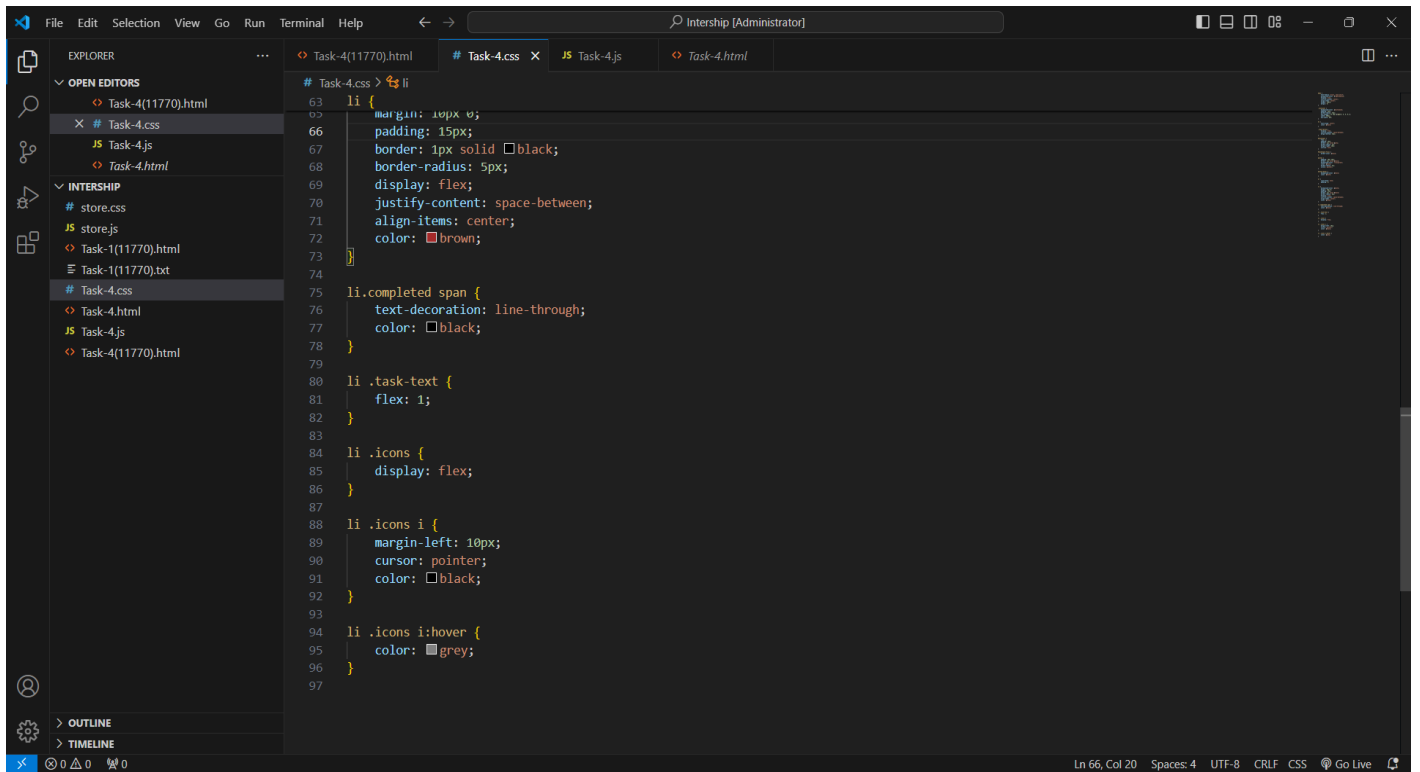
A screenshot of the Visual Studio Code editor interface. The Explorer panel on the left shows the file structure with 'Task-4(11770).html' selected. The main editor area displays the content of 'Task-4.css'. The code defines styles for the body, a container, an h1, an input group, a task input, and a button. The status bar at the bottom indicates 'Ln 66, Col 20'.

```
# Task-4.css >  li
1  body {
2      font-family: Arial, sans-serif;
3      background-color: lightskyblue;
4      display: flex;
5      justify-content: center;
6      align-items: center;
7      height: 100vh;
8      margin: 0;
9  }
10
11  .container {
12      background-color: whitesmoke;
13      padding: 20px;
14      border-radius: 10px;
15      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
16      width: 400px;
17      max-width: 80%;
18  }
19
20  h1 {
21      text-align: center;
22      color: black;
23  }
24
25  .input-group {
26      display: flex;
27      justify-content: space-between;
28      margin-bottom: 20px;
29  }
30
31  #taskInput {
32      flex: 1;
33      padding: 10px;
34      border: 1px solid grey;
35      border-radius: 5px;
36      margin-right: 10px;
37      outline: none;
```

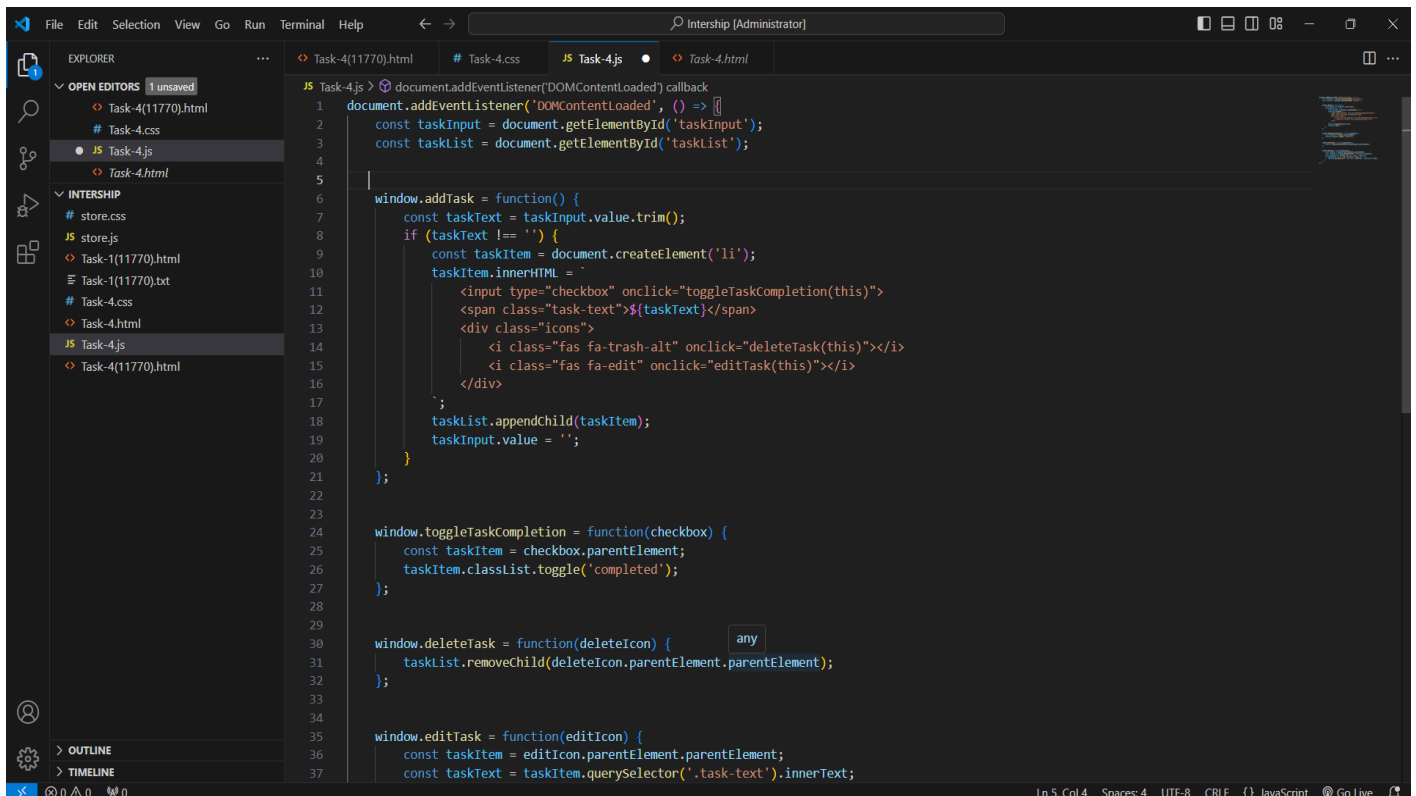


A screenshot of the Visual Studio Code editor interface, showing the continuation of the CSS code for 'Task-4.css'. The code defines styles for the task input focus state, a button, a button hover state, a list, and list items. The status bar at the bottom indicates 'Ln 66, Col 20'.

```
31  #taskInput {
32      flex: 1;
33      padding: 10px;
34      border: 1px solid grey;
35      border-radius: 5px;
36      margin-right: 10px;
37      outline: none;
38  }
39
40  #taskInput:focus {
41      border-color: black;
42  }
43
44  button {
45      padding: 10px 20px;
46      border: 1px solid black;
47      background-color: transparent;
48      color: black;
49      border-radius: 5px;
50      cursor: pointer;
51  }
52
53  button:hover {
54      background-color: black;
55      color: white;
56  }
57
58  ul {
59      list-style: none;
60      padding: 0;
61  }
62
63  li {
64      background-color: white;
65      margin: 10px 0;
66      padding: 15px;
67      border: 1px solid black;
68      border-radius: 5px;
69      display: flex;
70      justify-content: space-between;
```



.js



```
File Edit Selection View Go Run Terminal Help
Task-4(11770).html Task-4.css JS Task-4.js Task-4.html
OPEN EDITORS 1 unsaved
Task-4(11770).html
Task-4.css
JS Task-4.js
Task-4.html
INTERSHIP
store.css
store.js
Task-1(11770).html
Task-1(11770).txt
Task-4.css
Task-4.html
JS Task-4.js
Task-4(11770).html
OUTLINE
TIMELINE
Ln 5, Col 4 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live
```

JS Task-4.js > document.addEventListener('DOMContentLoaded') callback

```
1 document.addEventListener('DOMContentLoaded', () => {
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 window.toggleTaskCompletion = function(checkbox) {
26     const taskItem = checkbox.parentElement;
27     taskItem.classList.toggle('completed');
28 };
29
30 window.deleteTask = function(deleteIcon) {
31     taskList.removeChild(deleteIcon.parentElement.parentElement);
32 };
33
34
35 window.editTask = function(editIcon) {
36     const taskItem = editIcon.parentElement.parentElement;
37     const taskText = taskItem.querySelector('.task-text').innerText;
38     const newTaskText = prompt("Edit your task:", taskText);
39     if (newTaskText !== null && newTaskText.trim() !== '') {
40         taskItem.querySelector('.task-text').innerText = newTaskText.trim();
41     }
42 };
43
44 });
```