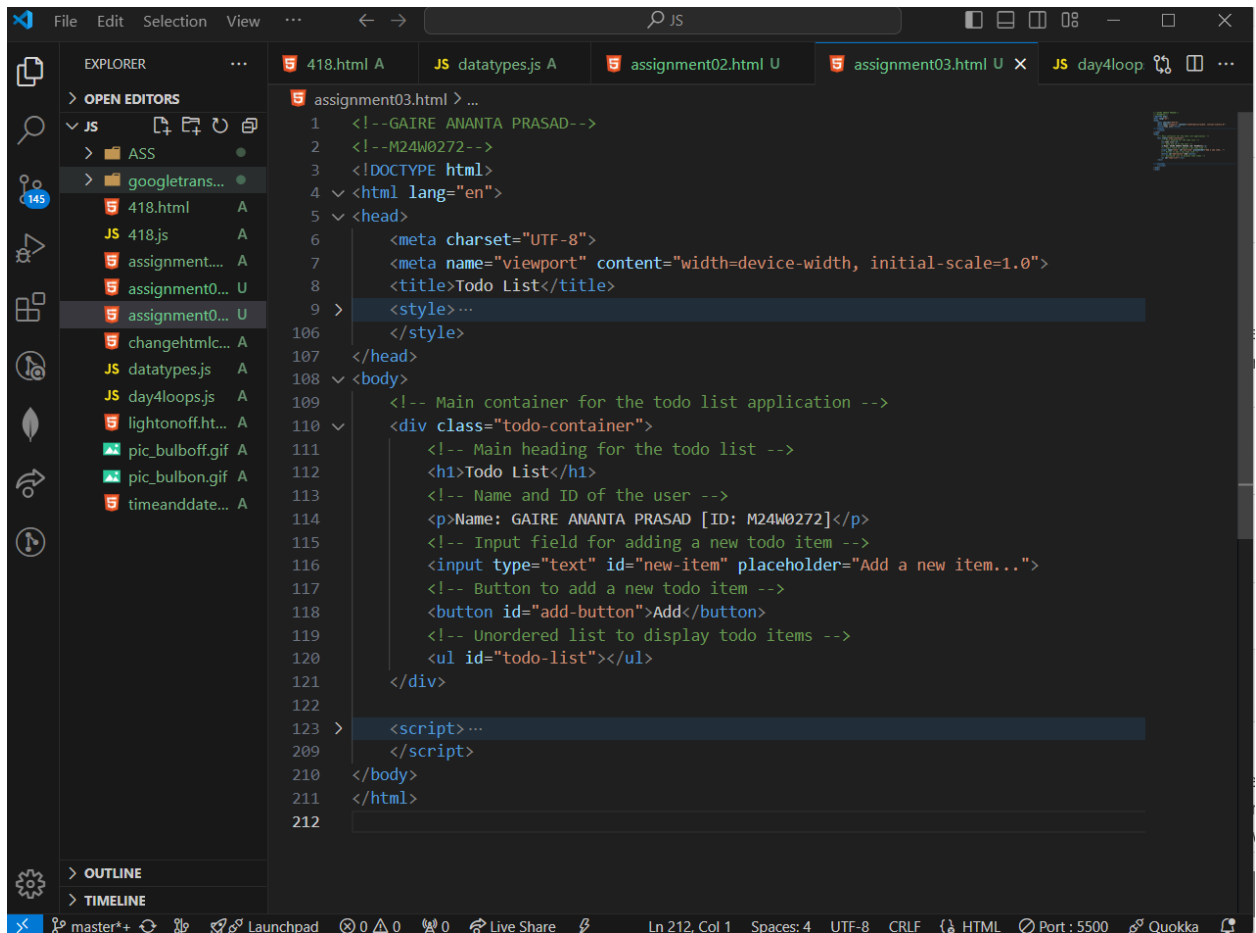


NAME: GAIRE ANANTA PRASAD

STUDENT ID: M24W0272

## 1. HTML5 CODE FOR TODO LIST.



The screenshot shows a VS Code editor with the following structure:

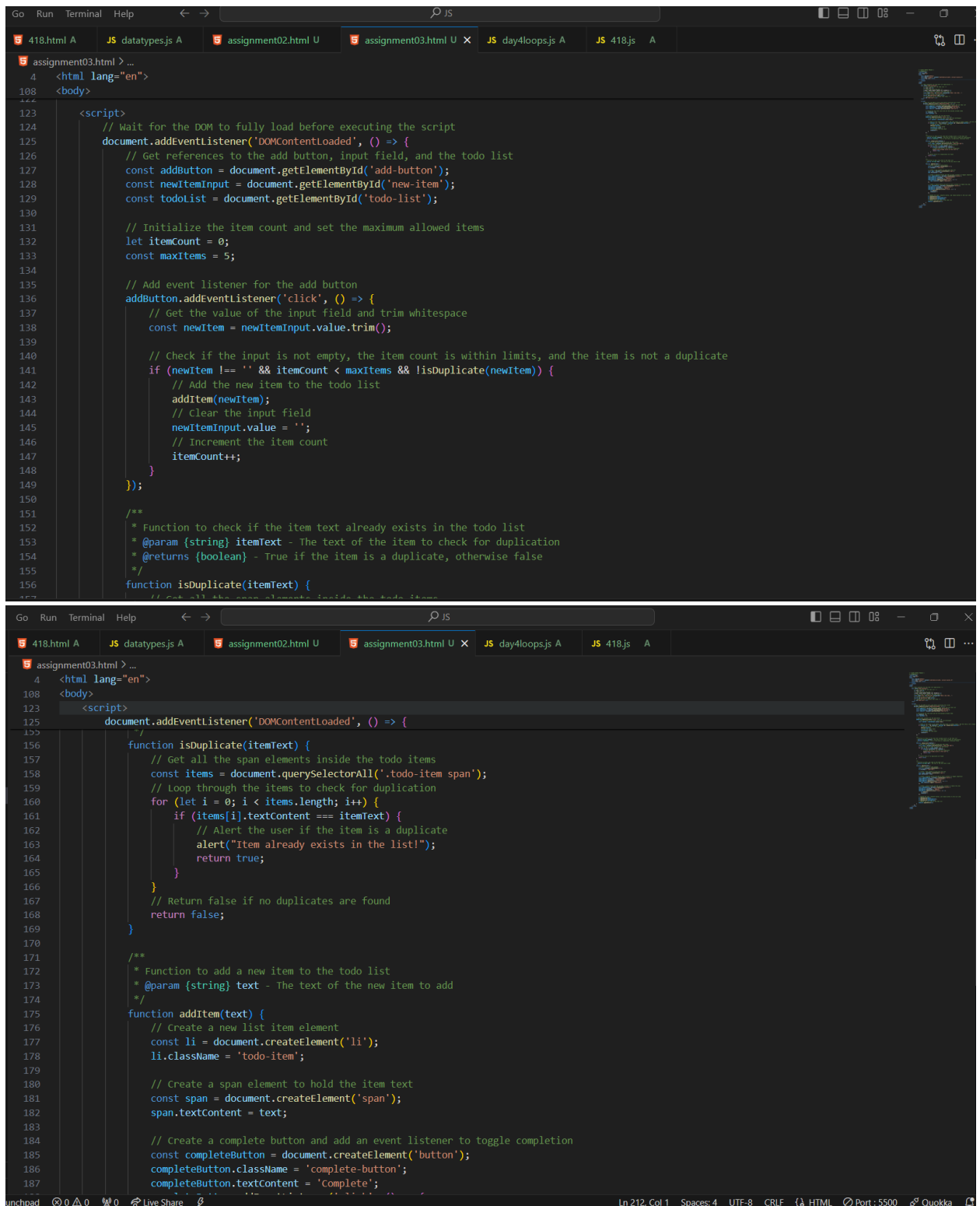
- EXPLORER:** Shows a file tree with folders 'ASS' and 'googletrans...'. Under 'ASS', there are files: '418.html', '418.js', 'assignment...', 'assignment0...', 'assignment0...', 'changehtmlc...', 'datatypes.js', 'day4loops.js', 'lighttonoff.ht...', 'pic\_bulboff.gif', 'pic\_bulbon.gif', and 'timeanddate...'. The 'assignment0...' file is selected.
- OPEN EDITORS:** Shows several open files, including 'assignment03.html U' which is the active file.
- assignment03.html > ...:** The main editor area displays the following HTML code:

```
1 <!-- GAIRE ANANTA PRASAD -->
2 <!-- M24W0272 -->
3 <!DOCTYPE html>
4 <html lang="en">
5 <head>
6   <meta charset="UTF-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Todo List</title>
9   <style>...
106 </style>
107 </head>
108 <body>
109   <!-- Main container for the todo list application -->
110   <div class="todo-container">
111     <!-- Main heading for the todo list -->
112     <h1>Todo List</h1>
113     <!-- Name and ID of the user -->
114     <p>Name: GAIRE ANANTA PRASAD [ID: M24W0272]</p>
115     <!-- Input field for adding a new todo item -->
116     <input type="text" id="new-item" placeholder="Add a new item...">
117     <!-- Button to add a new todo item -->
118     <button id="add-button">Add</button>
119     <!-- Unordered list to display todo items -->
120     <ul id="todo-list"></ul>
121   </div>
122
123   <script>...
209 </script>
210 </body>
211 </html>
212
```
- OUTLINE:** Shows the structure of the document, including 'OUTLINE' and 'TIMELINE'.
- STATUS BAR:** Shows the current file is 'master++', the editor is 'Launchpad', and the file is 'Ln 212, Col 1'. It also shows 'Spaces: 4', 'UTF-8', 'CRLF', and 'HTML'.

GAIRE ANANTA PRASAD

M24W0272

## 2. JAVASCRIPT FOR THE TODO LIST



```
Go Run Terminal Help JS
assignment03.html > ...
4 <html lang="en">
108 <body>
123 <script>
124 // Wait for the DOM to fully load before executing the script
125 document.addEventListener('DOMContentLoaded', () => {
126 // Get references to the add button, input field, and the todo list
127 const addButton = document.getElementById('add-button');
128 const newItemInput = document.getElementById('new-item');
129 const todoList = document.getElementById('todo-list');
130
131 // Initialize the item count and set the maximum allowed items
132 let itemCount = 0;
133 const maxItems = 5;
134
135 // Add event listener for the add button
136 addButton.addEventListener('click', () => {
137 // Get the value of the input field and trim whitespace
138 const newItem = newItemInput.value.trim();
139
140 // Check if the input is not empty, the item count is within limits, and the item is not a duplicate
141 if (newItem !== '' && itemCount < maxItems && !isDuplicate(newItem)) {
142 // Add the new item to the todo list
143 addItem(newItem);
144 // Clear the input field
145 newItemInput.value = '';
146 // Increment the item count
147 itemCount++;
148 }
149 });
150
151 /**
152 * Function to check if the item text already exists in the todo list
153 * @param {string} itemText - The text of the item to check for duplication
154 * @returns {boolean} - True if the item is a duplicate, otherwise false
155 */
156 function isDuplicate(itemText) {
157 // Get all the span elements inside the todo items
158 const items = document.querySelectorAll('li span');
159 // Loop through the items to check for duplication
160 for (let i = 0; i < items.length; i++) {
161 if (items[i].textContent === itemText) {
162 // Alert the user if the item is a duplicate
163 alert("Item already exists in the list!");
164 return true;
165 }
166 }
167 // Return false if no duplicates are found
168 return false;
169 }
170
171 /**
172 * Function to add a new item to the todo list
173 * @param {string} text - The text of the new item to add
174 */
175 function addItem(text) {
176 // Create a new list item element
177 const li = document.createElement('li');
178 li.className = 'todo-item';
179
180 // Create a span element to hold the item text
181 const span = document.createElement('span');
182 span.textContent = text;
183
184 // Create a complete button and add an event listener to toggle completion
185 const completeButton = document.createElement('button');
186 completeButton.className = 'complete-button';
187 completeButton.textContent = 'complete';
188 completeButton.addEventListener('click', () => {
189 // Toggle the completion status of the item
190 const item = li.querySelector('span');
191 if (item.classList.contains('complete')) {
192 item.classList.remove('complete');
193 } else {
194 item.classList.add('complete');
195 }
196 });
197 li.appendChild(span);
198 li.appendChild(completeButton);
199 todoList.appendChild(li);
200 }
```

```
418.html A JS datatypes.js A assignment02.html U assignment03.html U x JS day4loops.js A JS 418.js A
assignment03.html > ...
4 <html lang="en">
108 <body>
123 <script>
125   document.addEventListener('DOMContentLoaded', () => {
175     function addItem(text) {
184       // Create a complete button and add an event listener to toggle completion
185       const completeButton = document.createElement('button');
186       completeButton.className = 'complete-button';
187       completeButton.textContent = 'Complete';
188       completeButton.addEventListener('click', () => {
189         span.classList.toggle('completed');
190       });
191
192       // Create a remove button and add an event listener to remove the item
193       const removeButton = document.createElement('button');
194       removeButton.className = 'remove-button';
195       removeButton.textContent = 'Remove';
196       removeButton.addEventListener('click', () => {
197         li.remove();
198         itemCount--;
199       });
200
201       // Append the span, complete button, and remove button to the list item
202       li.appendChild(span);
203       li.appendChild(completeButton);
204       li.appendChild(removeButton);
205       // Append the list item to the todo list
206       todoList.appendChild(li);
207     }
208   });
209 </script>
210 </body>
211 </html>
212
```

### 3. CSS CODE FOR TODO LIST

```
418.html A JS datatypes.js A assignment02.html U assignment03.html U x JS day4loops.js A JS 418.js A
assignment03.html > ...
4 <html lang="en">
5 <head>
9 <style>
10   /* Styling for the body with a gradient background */
11   body {
12     font-family: Arial, sans-serif;
13     display: flex;
14     justify-content: center;
15     align-items: center;
16     height: 100vh;
17     margin: 0;
18     background: linear-gradient(135deg, #ffafbd, #ffcc3a0, #ffcc00);
19   }
20
21   /* Container styling with a gradient background, padding, and shadow */
22   .todo-container {
23     background: linear-gradient(135deg, #89fffd, #43e97b, #38f9d7);
24     padding: 20px;
25     border-radius: 15px;
26     box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);
27     width: 320px;
28     max-width: 90%;
29     text-align: center;
30   }
31
32   /* Style for the main heading */
33   h1 {
34     margin-top: 0;
35     color: #fff;
36   }
37
38   /* Style for the name paragraph */
39   p {
40     color: #fff;
41   }
42
43   /* Style for the input field */
```

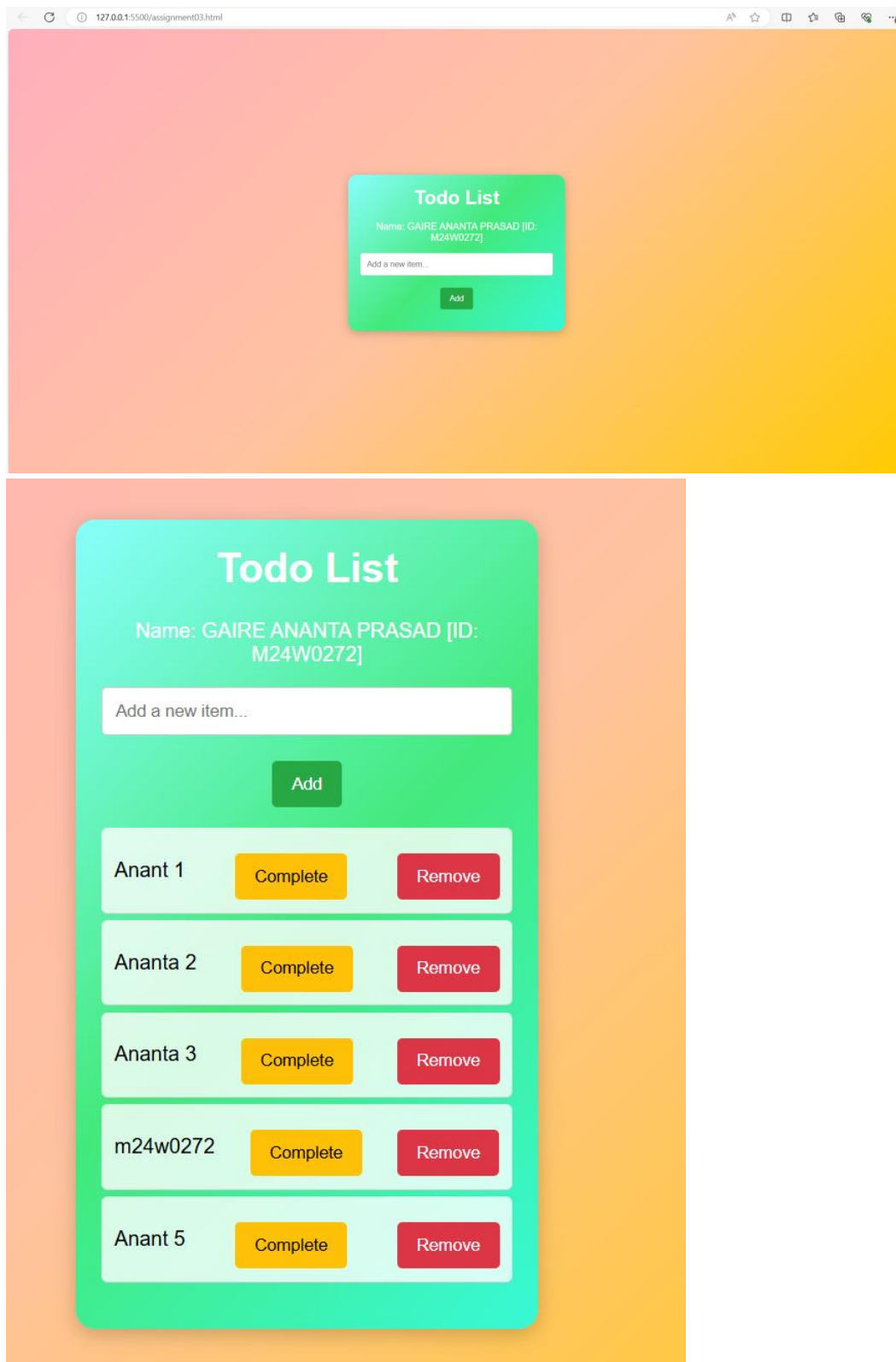
GAIRE ANANTA PRASAD  
M24W0272

```
418.html A JS datatypes.js A assignment02.html U assignment03.html U X JS day4loops.js A JS 418.js A
assignment03.html > ...
4 <html lang="en">
5 <head>
9 <style>
17
38 /* Style for the name paragraph */
39 p {
40   color: #fff;
41 }
42
43 /* Style for the input field */
44 input[type="text"] {
45   width: calc(100% - 22px);
46   padding: 10px;
47   margin-bottom: 10px;
48   border: 1px solid #ccc;
49   border-radius: 4px;
50 }
51
52 /* General button styling */
53 button {
54   padding: 10px 15px;
55   border: none;
56   border-radius: 4px;
57   cursor: pointer;
58   margin-top: 10px;
59 }
60
61 /* Specific styling for the Add button */
62 #add-button {
63   background-color: #28a745;
64   color: white;
65 }
66
67 /* Remove default list styling */
68 #todo-list {
69   list-style: none;
70   padding: 0;
71 }
72
```

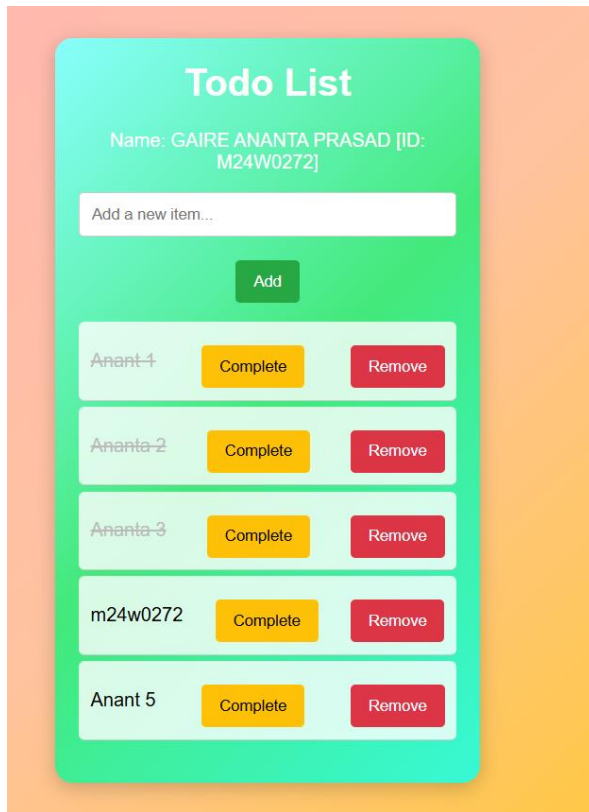
```
418.html A JS datatypes.js A assignment02.html U assignment03.html U X JS day4loops.js A JS 418.js A
assignment03.html > ...
4 <html lang="en">
5 <head>
9 <style>
65
66 /* Remove default list styling */
67 #todo-list {
68   list-style: none;
69   padding: 0;
70 }
71
72 /* Style for each todo item */
73 .todo-item {
74   display: flex;
75   justify-content: space-between;
76   align-items: center;
77   padding: 10px;
78   border-bottom: 1px solid #ccc;
79   background-color: rgba(255, 255, 255, 0.8);
80   border-radius: 5px;
81   margin-bottom: 5px;
82 }
83
84 /* Style for todo item buttons */
85 .todo-item button {
86   margin-left: 5px;
87 }
88
89 /* Specific styling for the Complete button */
90 .complete-button {
91   background-color: #ffc107;
92 }
93
94 /* Specific styling for the Remove button */
95 .remove-button {
96   background-color: #dc3545;
97   color: white;
98 }
99
100
101 /* Style for completed items */
102 .completed {
103   text-decoration: line-through;
104   color: #bbb;
105 }
106 </style>
107 </head>
108
```

GAIRE ANANTA PRASAD  
M24W0272

#### 4. OUTPUT FOR THE TODO LIST.

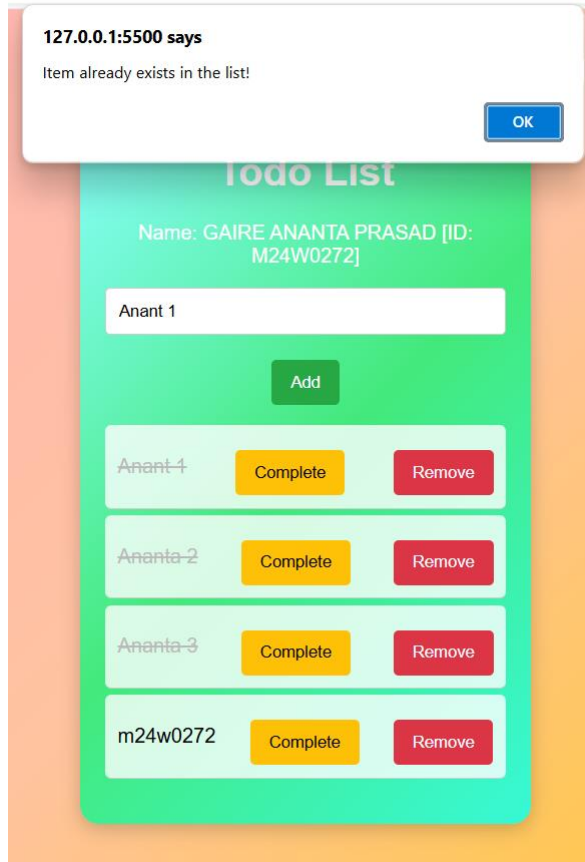


GAIRE ANANTA PRASAD  
M24W0272



same task is not accepted twice

GAIRE ANANTA PRASAD  
M24W0272



GAIRE ANANTA PRASAD  
M24W0272