

****Title: CodeTech IT Solutions Internship – Task Documentation: TO-DO LIST WEB APPLICATION****

****Introduction: ****

This documentation provides a detailed explanation of the task assigned during the CodeTech IT Solutions Internship program. The task involves a Web program for TO-DO List. This documentation will cover the implementation details, rationale behind the code structure, and insights into the programming techniques utilized. Additionally, it will include about the intern, Chandolu Sai amareshwar, and assigned ID, COD4451.

****Intern information: ****

Name: Chandolu Sai amareshwar

Intern ID: COD4451

****Task Description: ****

The task assigned to Chandolu Sai amareshwar during the CodeTech IT solutions internship program is to write a code using HTML, CSS and JavaScript for TO-DO List web application.

****Implementation: ****

The implementation of the task involves utilizing HTML, CSS and JavaScript to create a simple program that gives the output for TO-DO List. The program uses a HTML for providing the structure for the webpage, CSS for designing the webpage, JavaScript for providing the functionality to the webpage. Below is the code implementation.

...HTML

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>

<div id="myDIV" class="header">
  <h2 style="margin:5px">To Do List</h2>
  <input type="text" id="myInput" placeholder="Add things">
  <span onclick="newElement()" class="addBtn">Add</span>
</div>

<ul id="myUL">

</ul>
```

****Code Explanation: ****

HTML code is for creating the structure for the webpage.

<!DOCTYPE html> ==> This represents that the document type is HTML and the version is 5, It is an unpaired tag

<html> </html> ==> It is a paired tag; this represents root element of HTML. From here the HTML code will start.
</html> ==> This represents the close of the HTML code

<head>==> This represents opening of head tag, It is a paired tag

<title> ==> This represents the title of the webpage

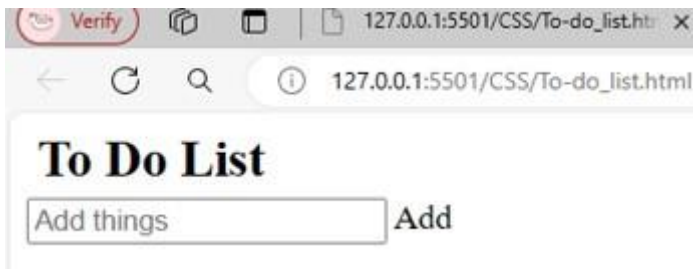
<body> ==> It is a opening of the body region. It acts as a container for the content which must be visible on the webpage, It is a paired tag.
</body> ==> It represents closing of the body.

<div id="myDIV" class="header">: This is a <div> element with the id "myDIV" and the class "header".
It's used to group the elements together. The class "header" suggests it's likely meant for styling purposes.

<h2 style="margin:5px">To Do List</h2>: This is a level 2 heading (<h2>) displaying the text "To Do List".
The inline style margin:5px adds a margin of 5 pixels around the heading.

<input type="text" id="myInput" placeholder="Add things">: This is an <input> element of type "text" with the id "myInput". It serves as a text field where users can input tasks to add to the to-do list. The placeholder text "Add things" suggests what users should input.

The output of the above code is shown below.



...CSS

```
<style> body{ margin: 0;
min-width: 250px;
}
```

```
*{ box-sizing: border-box;
}
```

```
ul{ margin: 0; padding: 0;
}
```

```
ul li{
cursor: pointer;
position: relative; padding: 12px 8px
12px 40px; list-style-type: none;
background: #eee; font-size: 18px;
transition: 0.2s;
```

```
-webkit-user-select: none;
-moz-user-select: none;
-ms-user-select: none;
user-select: none;
}
```

```
ul li:nth-child(odd) { background: white;
}
```

```
ul li.checked { background:gray;
color:white;
text-decoration: underline;
}
```

```
ul li.checked::before { content: "; position: absolute;
border-color:white;
border-style: solid;
border-width: 0 2px 2px 0;
top: 10px;
left: 16px;
transform: rotate(45deg);
height: 15px;
width: 7px;
}
```

```
.close {
position: absolute;
right: 0;
top: 0;
padding: 12px 16px 12px 16px;
}
```

```
.close:hover {
background-color:pink;
color: white;
}
```

```
.header {
background-color:blue; padding:
30px 40px; color: white;
text-align: center;
}
```

```
.header:after { content: "";
display: table; clear: both;
}
```

```
input {
margin: 0;
```

```
border: none;    border-  
radius: 0;    width: 75%;  
padding: 10px; float: left;  
font-size: 16px;  
}
```

```
.addBtn {  
padding: 10px; width: 25%;  
background: gray;    color:  
#b3a9a9; float: left;  
text-align: center;    font-  
size: 16px; cursor: pointer;  
transition: 0.3s;    border-  
radius: 0;  
}
```

```
.addBtn:hover {  
    background-color: #bbb;  
}  
</style>
```

****Code Explanation: ****

CSS is used to decorate the webpage ,to decorate the webpage Style tag is used,inside the style tag we have to give the requirements.

Body and General Styles:

- **margin: 0;** Removes default margin.
- **min-width: 250px;** Specifies the minimum width of the body.
- **box-sizing: border-box;** Ensures that padding and border are included in the element's total width and height.

Unordered List (ul):

- **margin: 0; padding: 0;** Removes default margin and padding.

List Items (ul li):

- **cursor: pointer;** Changes cursor to pointer when hovering over list items.
- **position: relative;** Allows positioning relative to the parent element.
- **padding: 12px 8px 12px 40px;** Sets padding for list items.
- **list-style-type: none;** Removes default list bullet points.
- **background: #eee;** Sets background color for list items.
- **font-size: 18px;** Sets font size for list items.
- **transition: 0.2s;** Adds a smooth transition effect.
- **user-select: none;** Prevents text selection.

Odd List Items (ul li:nth-child(odd)):

- **background: white;** Sets background color for odd list items.

Checked List Items (ul li.checked):

- **background: gray; color: white;;** Sets background and text color for checked items.
- **text-decoration: underline;;** Underlines checked items.

Checked List Item Before (ul li.checked::before):

- Adds a checkmark symbol before checked items using CSS pseudo-elements.

Close Button (.close):

- **position: absolute; right: 0; top: 0;;** Positions the close button to the top right corner of the list item.
- **padding: 12px 16px 12px 16px;;** Sets padding for the close button.
- **background-color: pink; color: white;;** Defines colors for the close button on hover.

Header (.header):

- **background-color: blue;;** Sets background color for the header.
- **padding: 30px 40px;;** Sets padding for the header.
- **color: white;;** Sets text color for the header.
- **text-align: center;;** Centers the text within the header.

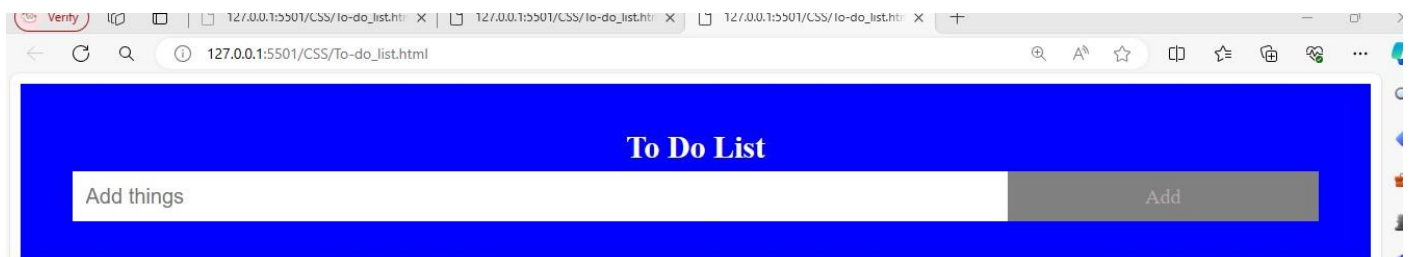
Input Field (input):

- **margin: 0; border: none; border-radius: 0;;** Resets default input styles.
- **width: 75%; padding: 10px; float: left;;** Sets width, padding, and float for the input field.
- **font-size: 16px;;** Sets font size for the input field.

Add Button (.addBtn):

- **padding: 10px; width: 25%;** Sets padding and width for the add button.
- **background: gray; color: #b3a9a9;;** Sets background and text color for the add button
- **cursor: pointer;;** Changes cursor to pointer when hovering over the add button.
- **transition: 0.3s;;** Adds a smooth transition effect for the add button.
- **border-radius: 0;;** Sets border-radius to 0 for the add button.

The output after writing the CSS Code is shown below.



...JavaScript

```
<script>
var myNodeList = document.getElementsByTagName("LI");
```

```

var i;
for (i = 0; i < myNodelist.length; i++) {
var span = document.createElement("SPAN");
var txt = document.createTextNode("\u00D7"); span.className = "close";
span.appendChild(txt);
myNodelist[i].appendChild(span);
}

```

```

var close = document.getElementsByClassName("close"); var i;
for (i = 0; i < close.length; i++)
{ close[i].onclick = function() {
var div = this.parentElement;
div.style.display = "none";
}
}

```

```

var list = document.querySelector('ul'); list.addEventListener('click',
function(ev) {
if(ev.target.tagName === 'LI') { ev.target.classList.toggle('checked');
}
}, false);

```

```

function newElement() {
var li = document.createElement("li");
var inputValue = document.getElementById("myInput").value; var t =
document.createTextNode(inputValue); li.appendChild(t);
if (inputValue === "") {
alert("You must write something!");
} else { document.getElementById("myUL").appendChild(li);
}
document.getElementById("myInput").value = "";
var span = document.createElement("SPAN"); var txt =
document.createTextNode("\u00D7"); span.className =
"close"; span.appendChild(txt);
li.appendChild(span);
for (i = 0; i < close.length; i++) { close[i].onclick =
function() { var div = this.parentElement;
div.style.display = "none";
}
}
}
</script>

```

****Code Explanation: ****

JavaScript code is for providing the functionality for the webpage.

Creating Close Buttons:

- The script starts by getting all the list items (**LI**) on the page.

- For each list item, it creates a **SPAN** element which will serve as a close button.
- The close button is represented by the Unicode character "\u00D7", which is the multiplication sign (×).
- The close button is given a class name "close" and appended to each list item.

Closing List Items:

- It then selects all elements with the class name "close".
- For each close button, an onclick event listener is attached.
- When a close button is clicked, it finds the parent element (the list item) and sets its display style to "none", effectively hiding it from view.

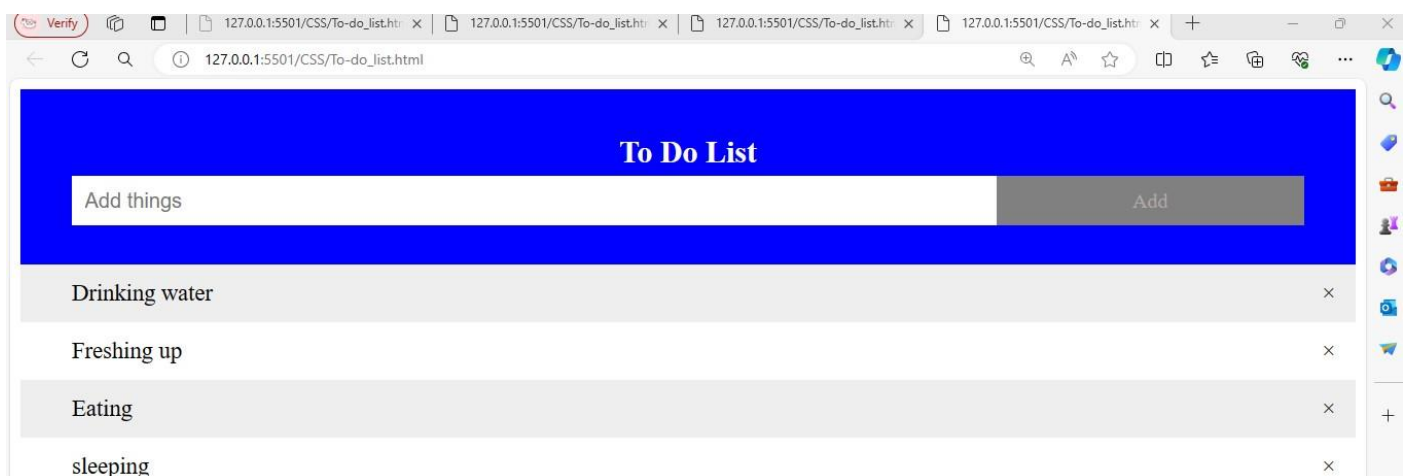
Checking and Unchecking Items:

- It selects the **ul** element (the parent of all list items) and adds a click event listener to it.
- When a click event occurs within the **ul** element, it checks if the clicked element is a list item (**LI**).
- If the clicked element is a list item, it toggles the class "checked" on that element. This class is used to visually mark items as checked.

Adding New Items:

- The **newElement()** function is defined, which is called when the user wants to add a new item.
- It starts by creating a new **LI** element and retrieving the value of the input field with the id "myInput".
- If the input field is empty, it displays an alert prompting the user to write something.
- If the input field is not empty, it creates a text node with the input value and appends it to the new **LI** element.
- The new **LI** element is then appended to the **ul** element with the id "myUL".
- After adding the new list item, it clears the input field.
- Finally, it creates a close button for the new list item in the same way as described earlier and attaches an onclick event listener to it for closing functionality.

The Final output after providing the javascript code is shown below



****Rationale: ****

The program utilizes the HTML code for providing the structure for the webpage, CSS Code provides the decoration for the web page for looking more attractive and JavaScript code provides the functionality for the web page so that it works effectively.

****Conclusion: ****

In conclusion, the task assigned to Chandolu Sai amareshwar during the CodeTech IT solutions internship program involved writing a code using Web technologies to provide a webpage for TO-DO List. The implemented solution successfully accomplishes this task using HTML, CSS and JavaScript. This documentation provides insights into implementation details, code explanation, and rationale behind the chosen approach. Chandolu Sai amareshwar with Intern ID COD4451, has effectively completed this task as part of the internship program.

This concludes the documentation for the task “TO-DO List Web application” assigned during the CodeTech IT Solutions internship program.