**TITLE:** CodTech IT Solutions Internship - Task Documentation: “To-DO LIST” Using CSS, HTML, JAVASCRIPT.

**INTERN INFORMATION:**

**Name:** MA Basith

**ID:** C0D4277

**HTML Structure:**

- `<!DOCTYPE html>`: Specifies the document type and version of HTML being used.

- `<html>`: The root element of the HTML document.

- `<head>`: Contains meta-information about the HTML document, such as the title, character encoding, and viewport settings.

- `<meta name="viewport" content="width=device-width,initial-scale=1">`: Sets the viewport to the width of the device's screen and sets the initial zoom level to 1.

- `<title>`: Sets the title of the HTML document.

- `<link rel="stylesheet" href="style.css">`: Links an external CSS file named "style.css" to the HTML document for styling.

- `<body>`: Contains the content of the HTML document visible to users.

- `<div class="container">`: Creates a container for the entire To-Do List application.

- `<div class="todo-app">`: Contains the To-Do List application.

- `<h2>`: Heading for the To-Do List.

- `<img src="Downloads/To-Do-Img/images/icon.png">`: Image tag with a source pointing to an icon image.

- `<div class="row">`: Creates a row for input elements.

- `<input type="text" id="input-box" placeholder="Add your text">`: Text input field for adding tasks.

- `<button onclick="addTask()">Add</button>`: Button to add a task, with an `onclick` attribute calling a JavaScript function named `addTask()`.

- `<ul id="list-container">`: Unordered list element to contain the list of tasks.

- List items (`<li>`) will be dynamically added here through JavaScript.

**CSS Structure:**

**1. Universal Selector (`\*`):**

- Sets margin, padding, font-family, and box-sizing to all elements. Resets default margin and padding to 0 and sets the font-family to 'Poppins' or sans-serif.

**2. Container Styles:**

- `.container`: Styles a container element.

- Sets width to 100%.

- Sets minimum height to 100vh (viewport height).

- Applies a linear gradient background color from #153677 to #4e085f.

- Adds padding of 10px.

**3. To-Do App Styles:**

- `.todo-app`: Styles the main to-do app container.

- Sets width to 100% but restricts maximum width to 540px.

- Applies a background color of white (#fff).

- Centers the container horizontally using auto margins and sets vertical margins.

- Adds padding to the top, right, and bottom, with extra padding at the bottom.

- Applies a border-radius of 10px for rounded corners.

**4. Header Styles:**

- `.todo-app h2`: Styles the heading of the to-do app.

- Sets color to a shade of blue (#002765).

- Aligns items along the center vertically using flexbox.

- Adds a margin at the bottom.

- `.todo-app h2 img`: Styles the image inside the heading.

- Sets width to 30px and adds a margin on the left.

**5. To-Do Item Styles:**

- `.row`: Styles each to-do item.

- Uses flexbox to align items and justify content.

- Sets a background color.

- Adds border-radius and padding.

- Adds margin at the bottom.

- `input`: Styles the input field inside each to-do item.

- Uses flex to occupy available space.

- Removes border and outline.

- Sets background and padding.

- Sets font weight.

- `button`: Styles the button inside each to-do item.

- Removes border and outline.

- Sets padding, background color, text color, font size, cursor, and border-radius.

**6. List Item Styles:**

- `ul li`: Styles the list items.

- Removes default list styling.

- Sets font size and padding.

- Prevents user selection.

- Adds cursor pointer.

- `ul li::before`: Styles the custom bullet point for list items.

- Uses absolute positioning to place it correctly.

- Sets size, border-radius, background image, and position.

- `ul li.checked`: Styles checked items.

- Sets color and text-decoration.

- `ul li.checked::before`: Styles the custom bullet point for checked items.

- Changes background image.

- `ul li span`: Styles the delete button for list items.

- Uses absolute positioning to place it correctly.

- Sets size, font size, color, line height, text alignment, and border-radius.

- `ul li span:hover`: Styles the delete button on hover.

- Changes background color.

**JavaScript:**

`<script src="script.js"></script>`: Links an external JavaScript file named "script.js" to the HTML document. This file likely contains the logic for adding tasks to the To-Do List.

**1. Variables:**

- `inputBox`: Stores a reference to the input element with the id "input-box".

- `listContainer`: Stores a reference to the container element where the list items will be appended, identified by the id "list-container".

**2. Function `addTask()`:**

- Checks if the value of the input box is empty.

- If empty, shows an alert indicating that something must be written.

- If not empty:

- Creates a new list item (`<li>`) element.

- Sets its inner HTML to the value of the input box.

- Appends this list item to the `listContainer`.

- Creates a delete button (a span with the text '×') for the list item.

- Appends this delete button to the list item.

- Resets the value of the input box to an empty string.

- Calls `saveData()` to save the updated list in local storage.

**3. Event Listener on `listContainer`:**

- Listens for clicks within the `listContainer`.

- If the clicked element is a list item (`<li>`):

- Toggles the "checked" class of the clicked list item to mark it as checked or unchecked.

- Calls `saveData()` to save the updated list in local storage.

- If the clicked element is a delete button (a `<span>` inside a list item):

- Removes the parent list item from the DOM.

- Calls `saveData()` to save the updated list in local storage.

**4. Function `saveData()`:**

- Saves the inner HTML content of the `listContainer` to the local storage with the key "data".

**5. Function `showTask()`:**

- Retrieves the saved list data from local storage with the key "data".

- Sets the inner HTML content of the `listContainer` to the retrieved data, effectively displaying the previously saved list items.

**6. Initial Execution:**

- Calls `showTask()` when the script runs to load and display the previously saved list items from local storage.

**CONCLUSION:**

This documentation provides an overview of the structure of the HTML document and its associated JavaScript and CSS files. For more detailed documentation on how the To-Do List functionality works.