**Q1: Can you create a simple HTML interface to add new items with fields like name and description?**

****

<!DOCTYPE html>

<html>

<head>

<title>Add New Item</title>

</head>

<body>

<h1>Add New Item</h1>

<form>

<label for="name">Name:</label>

<input type="text" id="name" name="name"><br><br>

<label for="description">Description:</label>

<textarea id="description" name="description"></textarea><br><br>

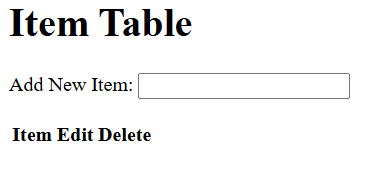
<input type="submit" value="Add Item">

</form>

</body>

</html>

**Q2: Build an HTML table with a form to add new items and buttons to edit or delete each one using JavaScript.**

****

<!DOCTYPE html>

<html>

<head>

<title>Item Table</title>

<script>

// function to handle form submission

function handleFormSubmit(event) {

event.preventDefault();

var newItem = document.getElementById('new-item').value;

var itemsTable = document.getElementById('items');

var newRow = itemsTable.rows[itemsTable.rows.length].cloneNode(true);

newRow.cells[0].innerHTML = newItem;

itemsTable.appendChild(newRow);

}

// function to edit an item

function handleEditEvent(event) {

event.preventDefault();

var row = event.target.parentNode.parentNode;

var itemName = row.cells[0].innerHTML;

var selectedRow = itemsTable.rows[itemsTable.rows.length - 1];

document.getElementById('edit-item').value = itemName;

}

// function to delete an item

function handleDeleteEvent(event) {

event.preventDefault();

var row = event.target.parentNode;

itemsTable.removeRow(row);

}

</script>

</head>

<body>

<h1>Item Table</h1>

<form id="item-form" onsubmit="handleFormSubmit(event)">

<label for="new-item">Add New Item:</label>

<input type="text" id="new-item" name="new-item"><br>

</form>

<table id="items">

<tr>

<th>Item</th>

<th>Edit</th>

<th>Delete</th>

</tr>

<!-- add new rows here -->

</table>

<!-- edit and delete buttons here -->

<script>

// bind events to buttons

document.getElementById('edit-btn').addEventListener('click', handleEditEvent);

document.getElementById('delete-btn').addEventListener('click', handleDeleteEvent);

// add new rows when button is clicked

document.getElementById('item-form').onsubmit = function() {

var newRow = itemsTable.rows[itemsTable.rows.length].cloneNode(true);

itemsTable.appendChild(newRow);

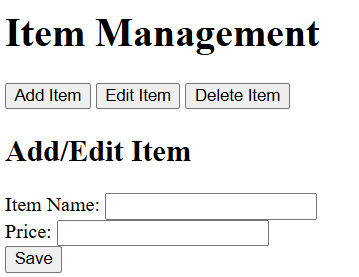
};

</script>

</body>

</html>

**Q3: Create a fully styled HTML and JavaScript interface for managing a list of items, including add, edit, and delete actions, with editable fields and confirmation prompts.**

****

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Item Management</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container">

<h1>Item Management</h1>

<div class="items-list">

<ul></ul>

<button id="add-item-btn">Add Item</button>

<button id="edit-item-btn">Edit Item</button>

<button id="delete-item-btn">Delete Item</button>

</div>

<div class="item-form">

<h2>Add/Edit Item</h2>

<label for="item-name">Item Name:</label>

<input type="text" id="item-name" name="item-name"><br>

<label for="item-price">Price:</label>

<input type="number" id="item-price" name="item-price"><br>

<button id="save-btn">Save</button>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

```

JavaScript Code:

```javascript

// Get the item list container

const itemsList = document.getElementById("items-list");

// Add item button click event listener

document.getElementById("add-item-btn").addEventListener("click", async () => {

// Create a new list item

const itemName = document.getElementById("item-name").value;

const itemPrice = document.getElementById("item-price").value;

// Add the item to the list

itemsList.innerHTML += `<li>${itemName} - ${itemPrice}</li>`;

document.getElementById("item-name").value = "";

document.getElementById("item-price").value = "";

});

// Edit item button click event listener

document.getElementById("edit-item-btn").addEventListener("click", async () => {

// Get the selected list item

const selectedItem = itemsList.querySelector(".selected");

// Create a new form to edit the item

const itemForm = document.createElement("div");

itemForm.classList.add("item-form");

// Add label and input fields for editing

const label = document.createElement("label");

label.innerHTML = `Item Name:`;

const inputName = document.createElement("input");

inputName.type = "text";

inputName.id = "item-name";

label.appendChild(inputName);

const labelPrice = document.createElement("label");

labelPrice.innerHTML = `Price:`;

const inputPrice = document.createElement("input");

inputPrice.type = "number";

inputPrice.id = "item-price";

labelPrice.appendChild(inputPrice);

// Add a save button

const saveBtn = document.createElement("button");

saveBtn.innerHTML = "Save";

saveBtn.addEventListener("click", async () => {

// Get the edited item name and price

const itemName = document.getElementById("item-name").value;

const itemPrice = document.getElementById("item-price").value;

// Add the edited item to the list

itemsList.innerHTML += `<li>${itemName} - ${itemPrice}</li>`;

document.getElementById("item-name").value = "";

document.getElementById("item-price").value = "";

});

// Append the form to the page

itemForm.appendChild(label);

itemForm.appendChild(inputName);

itemForm.appendChild(labelPrice);

itemForm.appendChild(inputPrice);

itemForm.appendChild(saveBtn);

document.body.appendChild(itemForm);

});

// Delete item button click event listener

document.getElementById("delete-item-btn").addEventListener("click", async () => {

// Get the selected list item

const selectedItem = itemsList.querySelector(".selected");

// Remove the selected item from the list

itemsList.innerHTML = itemsList.innerHTML.replace(

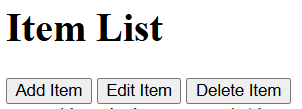
`<li class="selected">${selectedItem.textContent}</li>`,

""

);

});

**Consulta adicional: Create a fully styled HTML and JavaScript interface for managing a list of items, including add, edit, and delete actions, with editable fields and confirmation prompts. JavaScript must be placed inside a regular <script> tag — do not use DOMContentLoaded**

****

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Item List</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container">

<h1>Item List</h1>

<ul id="item-list"></ul>

<button id="add-item-btn">Add Item</button>

<button id="edit-item-btn">Edit Item</button>

<button id="delete-item-btn">Delete Item</button>

</div>

<script src="script.js"></script>

</body>

</html>

```

And here is the CSS:

```

\* {

box-sizing: border-box;

}

body {

font-family: Arial, sans-serif;

line-height: 1.6;

margin: 0;

padding: 20px;

}

#container {

display: flex;

flex-direction: column;

align-items: center;

gap: 20px;

}

h1 {

margin-bottom: 0;

}

ul {

list-style-type: none;

padding: 0;

margin: 20px 0;

}

#add-item-btn, #edit-item-btn, #delete-item-btn {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

}

#add-item-btn {

margin-bottom: 20px;

}

#edit-item-btn {

margin-bottom: 20px;

}

#delete-item-btn {

margin-bottom: 20px;

}

.form-control {

padding: 10px;

font-size: 16px;

border: none;

border-radius: 5px;

box-sizing: border-box;

}

.form-control:focus {

outline: none;

box-shadow: 0 0 0 2px #ccc;

}

```

And here is the JavaScript:

```

const itemList = document.getElementById("item-list");

const addItemButton = document.getElementById("add-item-btn");

const editItemButton = document.getElementById("edit-item-btn");

const deleteItemButton = document.getElementById("delete-item-btn");

let items = [];

function addItem() {

const itemName = prompt("Enter the name of the new item: ");

if (itemName) {

items.push({ name: itemName, id: items.length + 1 });

updateItemList();

}

}

function editItem(index) {

const selectedItem = items[index];

const updatedName = prompt("Enter the new name of the item: ");

if (updatedName) {

items[index].name = updatedName;

updateItemList();

}

}

function deleteItem(index) {

if (confirm("Are you sure you want to delete item #" + index + "?")) {

items.splice(index, 1);

updateItemList();

}

}

function updateItemList() {

const listItems = "";

items.forEach((item, index) => {

listItems += `<li id="${item.id}" class="item">${item.name}</li>`;

});

itemList.innerHTML = listItems;

}

addItemButton.addEventListener("click", addItem);

editItemButton.addEventListener("click", editItem);

deleteItemButton.addEventListener("click", deleteItem);

updateItemList();