

# Expensive Tracker

## HTML:

```
<!DOCTYPE html>
<html>

<head>
  <title>Expense Tracker</title>
  <link rel="stylesheet"
        type="text/css"
        href="expensive_tracker.css" />
  <link rel="stylesheet" href="expensive_tracker.js">
</head>

<body>
  <div class="container">
    <h1>Expense Tracker</h1>
    <form id="expense-form">
      <input type="text"
            id="expense-name"
            placeholder="Expense Name" required />
      <input type="number"
            id="expense-amount"
            placeholder="Amount" required />
      <button type="submit">
        Add Expense
      </button>
    </form>
    <div class="expense-table">
      <table>
        <thead>
          <tr>
            <th>Expense Name</th>
            <th>Amount</th>
            <th>Action</th>
          </tr>
        </thead>
        <tbody id="expense-list"></tbody>
      </table>
      <div class="total-amount">
        <strong>Total:</strong>
        $<span id="total-amount">0</span>
      </div>
    </div>
  </div>
  <script src="script.js"></script>
</body>
```

```
</html>
```

CSS:

```
/* style.css */
body {
  font-family: Arial, sans-serif;
  margin: 20px;
}

.container {
  max-width: 800px;
  margin: 0 auto;
  background-color: #f9f9f9;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

h1 {
  text-align: center;
  margin-bottom: 20px;
}

form {
  display: flex;
  justify-content: center;
  margin-bottom: 20px;
}

input[type="text"],
input[type="number"] {
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 4px;
  outline: none;
}

input[type="text"]::placeholder,
input[type="number"]::placeholder {
  color: #999;
}

button {
  padding: 10px 20px;
  background-color: #4caf50;
  color: white;
```

```

    border: none;
    border-radius: 4px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}

.expense-table {
    border: 1px solid #ddd;
    border-radius: 8px;
    overflow: hidden;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

table {
    width: 100%;
    border-collapse: collapse;
}

thead th {
    background-color: #f2f2f2;
    padding: 10px;
    text-align: left;
}

tbody td {
    padding: 10px;
    border-top: 1px solid #ddd;
}

.delete-btn {
    color: red;
    cursor: pointer;
}

.total-amount {
    padding: 10px;
    text-align: right;
    background-color: #f2f2f2;
}

```

JS:

```

// script.js
// Get form, expense list, and total amount elements
const expenseForm =
    document.getElementById("expense-form");

```

```
const expenselist =
  document.getElementById("expense-list");
const totalAmountElement =
  document.getElementById("total-amount");

// Initialize expenses array from localStorage
let expenses =
  JSON.parse(localStorage.getItem("expenses")) || [];

// Function to render expenses in tabular form
function renderExpenses() {

  // Clear expense list
  expenselist.innerHTML = "";

  // Initialize total amount
  let totalAmount = 0;

  // Loop through expenses array and create table rows
  for (let i = 0; i < expenses.length; i++) {
    const expense = expenses[i];
    const expenseRow = document.createElement("tr");
    expenseRow.innerHTML = `
      <td>${expense.name}</td>
      <td>${expense.amount}</td>
      <td class="delete-btn" data-id="${i}">Delete</td>
    `;
    expenselist.appendChild(expenseRow);

    // Update total amount
    totalAmount += expense.amount;
  }

  // Update total amount display
  totalAmountElement.textContent =
    totalAmount.toFixed(2);

  // Save expenses to localStorage
  localStorage.setItem("expenses",
    JSON.stringify(expenses));
}

// Function to add expense
function addExpense(event) {
  event.preventDefault();

  // Get expense name and amount from form
  const expenseNameInput =
```

```

        document.getElementById("expense-name");
const expenseAmountInput =
    document.getElementById("expense-amount");
const expenseName =
    expenseNameInput.value;
const expenseAmount =
    parseFloat(expenseAmountInput.value);

// Clear form inputs
expenseNameInput.value = "";
expenseAmountInput.value = "";

// Validate inputs
if (expenseName === "" || isNaN(expenseAmount)) {
    alert("Please enter valid expense details.");
    return;
}

// Create new expense object
const expense = {
    name: expenseName,
    amount: expenseAmount,
};

// Add expense to expenses array
expenses.push(expense);

// Render expenses
renderExpenses();
}

// Function to delete expense
function deleteExpense(event) {
    if (event.target.classList.contains("delete-btn")) {

        // Get expense index from data-id attribute
        const expenseIndex =
            parseInt(event.target.getAttribute("data-id"));

        // Remove expense from expenses array
        expenses.splice(expenseIndex, 1);

        // Render expenses
        renderExpenses();
    }
}

// Add event listeners

```

```
expenseForm.addEventListener("submit", addExpense);
expenseList.addEventListener("click", deleteExpense);

// Render initial expenses on page load
renderExpenses();
```

OUTPUT:

## Expense Tracker

Amount

Add Expense

Expense Name	Amount	Action
		Total: \$0