Solar Consumption Logic - JavaScript Steps

Step 1 - Data Model

Each consumption entry will follow this structure:

```
{
  id: 1,
  month: "January",
  kwh: 420
}

let consumptions = [];
```

Step 2 - Basic HTML Structure

```
<input type="text" id="month" placeholder="Month">
<input type="number" id="kwh" placeholder="kWh">
<button id="addConsumption">Add</button>

<div id="consumptionList"></div>
```

Step 3 - Add Entry Logic

```
const monthInput = document.getElementById("month");
const kwhInput = document.getElementById("kwh");
const addBtn = document.getElementById("addConsumption");
const listContainer = document.getElementById("consumptionList");

let consumptions = [];

addBtn.addEventListener("click", () => {
  const month = monthInput.value;
  const kwh = parseFloat(kwhInput.value);

  if (!month || !kwh) return;

  const newEntry = {
    id: Date.now(),
    month,
    kwh
  };

  consumptions.push(newEntry);
  renderConsumptions();

  monthInput.value = "";
```

```
  kwhInput.value = "";
});
```

Step 4 - Dynamic Rendering

```
function renderConsumptions() {
  listContainer.innerHTML = "";

  consumptions.forEach(entry => {
    const div = document.createElement("div");

    div.innerHTML = `
      <strong>${entry.month}</strong> - ${entry.kwh} kWh
    `;

    listContainer.appendChild(div);
  });
}
```

Bonus - Calculate Total Consumption

```
function calculateTotals() {
  const total = consumptions.reduce((sum, entry) => sum + entry.kwh, 0);
  return total;
}
```