<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Charging Bunk Simulator</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            padding: 20px;

        }

        .bunk {

            display: flex;

            gap: 10px;

            margin-top: 20px;

        }

        .slot {

            border: 1px solid #ccc;

            padding: 10px;

            width: 150px;

            height: 120px;

            text-align: center;

            border-radius: 8px;

            background-color: #f9f9f9;

        }

.occupied {

            background-color: #c3f3c3;

        }

        .available-slots {

            margin-bottom: 10px;

        }

    </style>

</head>

<body>

    <h2>Charging Bunk Simulator</h2>

    <label>Scooter ID: <input type="text" id="scooterId"></label>

    <label>Battery Level (%): <input type="number" id="batteryLevel" min="0" max="100"></label>

    <button onclick="checkIn()">Check In</button>

    <button onclick="simulateCharging()">Simulate Charging</button>

    <div class="available-slots" id="availableSlots"></div>

    <div class="bunk" id="bunk"></div>

    <script>

        const TOTAL\_SLOTS = 3;

        const slots = Array(TOTAL\_SLOTS).fill(null);

        function renderBunk() {

            const bunk = document.getElementById('bunk');

            bunk.innerHTML = '';

            const availableSlotsElement = document.getElementById('availableSlots');

            let availableSlotsCount = 0;

            for (let i = 0; i < TOTAL\_SLOTS; i++) {

                const slot = document.createElement('div');

                slot.className = 'slot';

                if (slots[i]) {

                    slot.classList.add('occupied');

                    slot.innerHTML = `

                        <strong>${slots[i].id}</strong><br>

                        Battery: ${slots[i].battery}%<br>

                        <button onclick="checkOut</button>

                    `;

                } else {

                    availableSlotsCount++;

                    slot.textContent = `Slot ${i + 1} - Empty`;

                }

                bunk.appendChild(slot);

            }

 availableSlotsElement.textContent = `Available Slots: ${availableSlotsCount}`;

        }

        function checkIn() {

            const id = document.getElementById('scooterId').value.trim();

            const battery = parseInt(document.getElementById('batteryLevel').value);

            if (!id || isNaN(battery) || battery < 0 || battery > 100) {

                alert("Please enter valid ID and battery level (0–100).");

                return;

            }

            const availableSlot = slots.findIndex(slot => slot === null);

            if (availableSlot === -1) {

                alert("No available slots.");

                return;

            }

            slots[availableSlot] = { id, battery };

            renderBunk();

        }

        function checkOut(index) {

            if (slots[index]) {

                alert(`Scooter ${slots[index].id} removed. Final Battery: ${slots[index].battery}%`);

                slots[index] = null;

                renderBunk();

            }

        }

function simulateCharging() {

            for (let i = 0; i < TOTAL\_SLOTS; i++) {

                if (slots[i]) {

                    slots[i].battery = Math.min(100, slots[i].battery + 10);

                }

            }

            renderBunk();

        }

        renderBunk();

    </script>

</body>

</html>

const TOTAL\_SLOTS = 3;

function renderBunk() {

  const bunk = document.getElementById("bunk");

  bunk.innerHTML = "";

  db.collection("scooters").orderBy("timestamp").limit(TOTAL\_SLOTS).get().then(snapshot => {

    snapshot.forEach(doc => {

      const data = doc.data();

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

      div.style.border = "1px solid #ccc";

      div.style.padding = "10px";

      div.style.margin = "10px";

      div.style.background = data.battery >= 100 ? "#c8f8c8" : "#f9f9f9";

      div.innerHTML = `

        <strong>${data.id}</strong><br>

        Battery: ${data.battery}%<br>

        <button onclick="checkOut('${doc.id}')">Check Out</button>

      `;

      bunk.appendChild(div);

    });

  });

}

function checkIn() {

  const id = document.getElementById("scooterId").value;

  const battery = parseInt(document.getElementById("batteryLevel").value);

  if (!id || battery < 0 || battery > 100) {

    alert("Invalid input");

    return;

  }

  db.collection("scooters").add({

    id,

    battery,

    timestamp: firebase.firestore.FieldValue.serverTimestamp()

  }).then(() => {

    updateAssistant(`Scooter ${id} added for charging.`);

    renderBunk();

  });

}

function checkOut(docId) {

  db.collection("scooters").doc(docId).delete().then(() => {

    updateAssistant("Scooter checked out.");

    renderBunk();

  });

}

function simulateCharging() {

  db.collection("scooters").get().then(snapshot => {

    snapshot.forEach(doc => {

      const data = doc.data();

      const newBattery = Math.min(100, data.battery + 10);

      db.collection("scooters").doc(doc.id).update({ battery: newBattery });

    });

    updateAssistant("All scooters charged +10%");

    renderBunk();

  });

}

function updateAssistant(msg) {

  document.getElementById("assistant").textContent = "🤖 Assistant: " + msg;

}

renderBunk();