```
<body>
<div class="wrap">
  <h1>Dose & Vial Optimizer — Fixed 70 mg / 100 mg</h1>
  <div class="sub">For HCPs: choose the dose regimen, enter weight, and get the vial counts that meet or exceed the required dose with minimum waste. <strong>Only 70 mg and 100 mg vials are available.</strong></div>

  <div class="banner">For HCP guidance only. Verify calculations against the product label and local policy. <strong>No underdosing</strong> and <strong>single-patient use</strong> are assumed. Required mg is rounded <strong>up</strong> to the nearest whole mg.</div>

  <div class="card" aria-labelledby="inputs-title">
    <h2 id="inputs-title">Inputs</h2>
    <div class="grid">
      <div>
        <label for="weight">Patient weight (kg)</label>
        <input id="weight" type="number" min="0" step="0.1" placeholder="e.g., 70"/>
      </div>
      <div>
        <label>Dose regimen (mg/kg)</label>
        <div class="row" role="radiogroup" aria-label="Dose regimen">
          <label><input type="radio" name="regimen" value="2.5" checked/> 2.5 mg/kg (Start)</label>
          <label><input type="radio" name="regimen" value="1.9"/> 1.9 mg/kg (Reduced)</label>
        </div>
      </div>
      <div>
        <label>Vial strengths (fixed)</label>
        <div class="row"><span class="pill" aria-label="Fixed vial strengths">70 mg</span><span class="pill" aria-hidden="true">100 mg</span></div>
      </div>
      <div class="no-print">
        <label>Max total vials</label>
        <input id="maxVials" type="number" min="1" step="1" value="10"/>
      </div>
    </div>
    <div class="row action-row">
      <div class="error" id="error" role="alert"></div>
      <div class="button-group no-print">
        <button class="btn" id="calcBtn" aria-label="Calculate options">Calculate options</button>
        <button class="btn secondary" id="resetBtn" aria-label="Reset">Reset</button>
```

```
      <button class="btn secondary" id="printBtn" aria-label="Print or Save as PDF">Print / Save PDF</button>
    </div>
  </div>
 </div>

 <div class="card">
  <div class="status-row">
    <div id="summary" class="info">Enter weight and select a regimen, then calculate.</div>
    <div id="requiredbox" class="info" style="display:none;"><strong>Total required dose:</strong> <span
id="requiredValue"></span> mg</div>
    <div id="recommend" class="rec" style="display:none;">Recommendation will appear here</div>
  </div>

  <div class="table-wrap">
    <table id="resultsTbl" style="display:none;">
      <thead>
        <tr>
          <th>Option</th>
          <th>70 mg vials</th>
          <th>100 mg vials</th>
          <th>Total (mg)</th>
          <th>Waste (mg)</th>
          <th>Waste (%)</th>
          <th>Total vials</th>
        </tr>
      </thead>
      <tbody id="resultsBody"></tbody>
    </table>
  </div>
  <div class="muted footnote">Tie-breakers (in order): least waste → fewest vials.</div>
 </div>

 <div class="card no-print" id="tests">
  <h3>Self-tests</h3>
  <div class="muted" id="testResults">Will run automatically…</div>
 </div>
</div>

<script>
 const $ = (id) => document.getElementById(id);
```

```
  const VIAL_A = 70; // mg
  const VIAL_B = 100; // mg

  // Actions
  $('printBtn')?.addEventListener('click', () => window.print());
  $('resetBtn')?.addEventListener('click', () => {
    $('weight').value=''; $('maxVials').value=10;
document.querySelector('input[name="regimen"][value="2.5"]').checked=true;
    $('resultsTbl').style.display='none'; $('resultsBody').innerHTML='';
    $('summary').textContent='Enter weight and select a regimen, then calculate.';
    $('recommend').style.display='none'; $('error').textContent='';
    $('requiredbox').style.display='none'; $('requiredValue').textContent='';
  });

  // Utils
  function fmt(n, d=0){ if(n==null||isNaN(n)) return ''; return
Number(n).toLocaleString(undefined,{maximumFractionDigits:d, minimumFractionDigits:d}); }
  function requiredMg(weight, mgPerKg){ return Math.ceil(weight * mgPerKg); }

  // Core enumeration (no underdosing, fixed 70/100 mg)
  function enumerate(required, maxVials){
    const out=[]; const a=VIAL_A, b=VIAL_B;
    const maxA = Math.min(maxVials, Math.ceil(required/a)+6);
    const maxB = Math.min(maxVials, Math.ceil(required/b)+6);
    for(let x=0;x<=maxA;x++){
      for(let y=0;y<=maxB;y++){
        const v=x+y; if(v===0||v>maxVials) continue;
        const total = x*a + y*b; if(total < required) continue; // no underdosing
        const waste = total - required; const wastePct = total>0? waste/total : 0;
        out.push({x,y,total,waste,wastePct,totalVials:v});
      }
    }
    out.sort((p,q)=>{
      if(p.waste!==q.waste) return p.waste-q.waste;
      if(p.totalVials!==q.totalVials) return p.totalVials-q.totalVials;
      return p.total-q.total;
    });
    if(out.length){
      const minWaste=out[0].waste; const minVAmong=
Math.min(...out.filter(r=>r.waste===minWaste).map(r=>r.totalVials));
```

```
    out.forEach(r=>{ if(r.waste===minWaste) r.leastWaste=true; if(r.waste===minWaste &&
r.totalVials===minVAmong) r.fewestVialsAmongLeast=true; });
  }
  return out;
 }

 function recommendation(opt){
  if(!opt) return '';
  const head = opt.waste===0? 'Exact match' : 'Min waste option';
  return `${head} → order ${opt.x}×70 mg and ${opt.y}×100 mg (Total ${fmt(opt.total,2)} mg; Waste
${fmt(opt.waste,2)} mg / ${(opt.wastePct*100).toFixed(2)}%; ${opt.totalVials} vial${opt.totalVials>1?'s':''})`;
 }

 // Calculate & render (max 4 options; hide dominated)
 $('calcBtn')?.addEventListener('click', () => {
  $('error').textContent=''; $('resultsTbl').style.display='none'; $('resultsBody').innerHTML='';
  const weight = parseFloat($('weight').value);
  const mgPerKg = parseFloat(document.querySelector('input[name="regimen"]:checked')?.value || '2.5');
  const maxVials = parseInt($('maxVials').value||'10',10);
  if(!(weight>0)) { $('error').textContent='Please enter patient weight.'; return; }

  const req = requiredMg(weight, mgPerKg);
  $('requiredValue').textContent = fmt(req,2);
  $('requiredbox').style.display = '';

  const opts = enumerate(req, maxVials);
  $('summary').textContent = `${fmt(mgPerKg,1)} mg/kg × ${fmt(weight,1)} kg → required ${fmt(req,2)} mg
(rounded up). Two vial strengths: 70 mg & 100 mg.`;

  if(!opts.length){ $('error').textContent='No valid combinations found with current settings. Try increasing max
vials.'; $('recommend').style.display='none'; return; }

  // Pareto filter (remove dominated by waste & vials)
  const pareto = opts.filter((p,i)=> !opts.some((q,j)=> j!==i && q.waste<=p.waste && q.totalVials<=p.totalVials &&
(q.waste<p.waste || q.totalVials<p.totalVials)));

  const best = pareto[0];
  const only70 = opts.find(o=> o.y===0);
  const only100 = opts.find(o=> o.x===0);
  const nextMixed = pareto.find(o=> o!==best && o.x>0 && o.y>0);
```

```
    let shown = [best, only70, only100, nextMixed].filter(Boolean);
    const uniq = []; const seen = new Set();
    for(const o of shown){ const k = `${o.x}|${o.y}`; if(!seen.has(k)){ seen.add(k); uniq.push(o);} }
    shown = uniq.slice(0,4);

    $('recommend').textContent = recommendation(best); $('recommend').style.display='';

    const tbody = $('resultsBody');
    shown.forEach((o,i)=>{
      const tr=document.createElement('tr');
      const n=document.createElement('td'); n.textContent=`Option ${i+1}`;
      if(o.waste===best.waste) { const b=document.createElement('span'); b.className='badge least';
b.textContent='Least waste'; n.appendChild(b);}
      if(o.totalVials===best.totalVials && o.waste===best.waste) { const b=document.createElement('span');
b.className='badge fewest'; b.textContent='Fewest vials'; n.appendChild(b);}
      tr.appendChild(n);
      const tdA=document.createElement('td'); tdA.textContent=fmt(o.x); tr.appendChild(tdA);
      const tdB=document.createElement('td'); tdB.textContent=fmt(o.y); tr.appendChild(tdB);
      const tdT=document.createElement('td'); tdT.textContent=fmt(o.total,2); tr.appendChild(tdT);
      const tdW=document.createElement('td'); tdW.textContent=fmt(o.waste,2); tr.appendChild(tdW);
      const tdWp=document.createElement('td'); tdWp.textContent=(o.total>0)?(o.wastePct*100).toFixed(2)+'%':'';
tr.appendChild(tdWp);
      const tdV=document.createElement('td'); tdV.textContent=fmt(o.totalVials); tr.appendChild(tdV);
      tbody.appendChild(tr);
    });
    $('resultsTbl').style.display='';
  });

  // ----------------------
  // Self-tests (simple, inline)
  // ----------------------
  function bestOptionFor(weight, mgPerKg){
    const req = requiredMg(weight, mgPerKg);
    const opts = enumerate(req, 10);
    return opts[0];
  }

  function runSelfTests(){
    const cases = [
```

```
    {name:'70 kg @ 2.5 mg/kg → 175 mg', w:70, d:2.5, expect:{x:0,y:2,waste:25}},
    {name:'70 kg @ 1.9 mg/kg → 133 mg', w:70, d:1.9, expect:{x:2,y:0,waste:7}},
    {name:'90 kg @ 2.5 mg/kg → 225 mg (mix allowed)', w:90, d:2.5, expect:{x:1,y:2,waste:45}},
    {name:'90 kg @ 1.9 mg/kg → 171 mg', w:90, d:1.9, expect:{x:0,y:2,waste:29}},
  ];
  const lines=[]; let pass=0, fail=0;
  for(const tc of cases){
    const b = bestOptionFor(tc.w, tc.d);
    const ok = b && b.x===tc.expect.x && b.y===tc.expect.y && b.waste===tc.expect.waste;
    if(ok){ pass++; lines.push(`✅ <span class="pass">PASS</span> — ${tc.name} → ${b.x}×70 + ${b.y}×100
(waste ${b.waste} mg)`); }
    else { fail++; lines.push(`❌ <span class="fail">FAIL</span> — ${tc.name}: expected ${tc.expect.x}×70 +
${tc.expect.y}×100 (waste ${tc.expect.waste}), got ${b?`${b.x}×70 + ${b.y}×100 (waste ${b.waste})`:'no option'}`);
}
  }
  $('testResults').innerHTML = `<strong>${pass} passed, ${fail} failed</strong><br>`+lines.join('<br>');
 }

 document.addEventListener('DOMContentLoaded', runSelfTests);
</script>
```