```
<!doctype html>

<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>Student Seeding Checker</title>
  <!-- Tailwind CDN for quick styling -->
  <script src="https://cdn.tailwindcss.com"></script>
  <style>
    /* Small custom styles */
    .glow { box-shadow: 0 6px 18px rgba(99,102,241,0.15); }
    .fade-in { animation: fadeIn .28s ease both; }
    @keyframes fadeIn { from { opacity: 0; transform: translateY(6px);} to { opacity:1; transform:
translateY(0);} }
    .status-pill { display:inline-flex; align-items:center; gap:8px; padding:6px 10px;
border-radius:9999px; font-weight:600; }
  </style>
</head>
<body class="bg-slate-50 text-slate-800 min-h-screen flex items-center justify-center p-6">
  <main class="max-w-4xl w-full">
    <section class="bg-white rounded-2xl p-8 shadow-md glow">
      <header class="flex items-start justify-between gap-4">
        <div>
          <h1 class="text-2xl md:text-3xl font-extrabold">Student Seeding Status — Instant
Checker</h1>
          <p class="mt-1 text-sm text-slate-500">Enter a student ID or email to instantly view
seeding status and actionable instructions to fix common issues.</p>
        </div>
        <div class="text-xs text-slate-400">Demo (front-end only)</div>
      </header><form id="checkForm" class="mt-6 grid grid-cols-1 md:grid-cols-3 gap-3
items-end">
    <div class="md:col-span-2">
      <label for="identifier" class="block text-sm font-medium text-slate-600">Student ID or
Email</label>
      <input id="identifier" type="text" placeholder="e.g. S202500123 or student@example.edu"
required
             class="mt-1 block w-full rounded-lg border-slate-200 shadow-sm
focus:border-indigo-500 focus:ring-indigo-400 p-3" />
    </div>

    <div class="flex gap-2">
      <button type="submit" class="w-full inline-flex justify-center items-center gap-2 rounded-lg
bg-indigo-600 text-white px-4 py-3 font-semibold hover:bg-indigo-700">
```

```
        Check Status

span </button> <button id="sampleBtn" type="button" class="w-full inline-flex justify-center
items-center gap-2 rounded-lg border border-slate-200 bg-white px-4 py-3 text-sm"> Use
Sample </button> </div> </form>

<div id="result" class="mt-6 fade-in hidden bg-slate-50 rounded-lg p-5 border border-slate-100">
    <!-- Dynamic content injected here -->
  </div>

  <hr class="my-6" />

  <details class="group p-4 rounded-lg bg-white border border-slate-100">
    <summary class="cursor-pointer font-semibold">How this works (front-end
demo)</summary>
    <div class="mt-3 text-sm text-slate-600">
      <p>This page simulates an instant lookup. In a real deployment you would POST the
identifier to your backend API; the backend would then query the seeding system (database /
payment gateway / bank API) and return a status and reason codes. This demo contains
sample statuses and step-by-step instructions for common failure modes.</p>
    </div>
  </details>
</section>

<section class="mt-6 text-sm text-slate-500">
  <p><strong>Note:</strong> This is a single-file front-end demo. If you want, I can scaffold a
small Node/Express endpoint and a simple JSON data store for prototyping the server side.</p>
</section>

  </main>  <script>
  // ---------------------------
  // Demo logic: replace with real API calls in production
  // ---------------------------

  const sampleDatabase = {
    // Sample responses keyed by identifier (case-insensitive)
    "s2025001": { status: 'seeded', lastUpdated: '2025-10-20', details: { program: 'B.Tech
Computer Science', bank: 'Seeded to A/C ****1234' } },
    "s2025002": { status: 'pending', lastUpdated: '2025-10-25', details: { reason: 'Awaiting bank
confirmation' } },
    "s2025003": { status: 'failed', lastUpdated: '2025-10-21', details: { code: 'MISSING_DOCS',
missing: ['Bank IFSC proof', 'Student Aadhaar'], hint:'Upload missing documents to student
portal -> Documents' } },
    "student@example.edu": { status: 'failed', lastUpdated: '2025-10-22', details: { code:
'BANK_MISMATCH', bankOnFile: 'HDFC ****2345', providedBank: 'SBI ****9999', hint: 'Update
bank details from Profile > Bank Details or contact accounts@college.edu' } }
  };

  const checkForm = document.getElementById('checkForm');
  const identifierInput = document.getElementById('identifier');
  const resultBox = document.getElementById('result');
  const sampleBtn = document.getElementById('sampleBtn');

  function renderStatusCard(record, query) {
    // Build UI elements
    resultBox.innerHTML = '';
    resultBox.classList.remove('hidden');

    const title = document.createElement('div');
    title.className = 'flex items-center justify-between gap-4';

    const left = document.createElement('div');
    left.innerHTML = `<div class="text-sm text-slate-500">Identifier</div><div class="text-lg
font-semibold">${escapeHtml(query)}</div>`;

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

    const status = (record && record.status) ? record.status : 'unknown';
    let pill = document.createElement('span');
    pill.className = 'status-pill';

    if (status === 'seeded') {
      pill.style.background = '#ecfdf5';
      pill.style.color = '#065f46';
      pill.textContent = 'Seeded ✓';
    } else if (status === 'pending') {
      pill.style.background = '#fff7ed';
      pill.style.color = '#92400e';
      pill.textContent = 'Pending…';
    } else if (status === 'failed') {
      pill.style.background = '#ffefef';
      pill.style.color = '#7f1d1d';
      pill.textContent = 'Failed ✕';
    } else {
      pill.style.background = '#eef2ff';
      pill.style.color = '#3730a3';
      pill.textContent = 'Unknown';
    }

    right.appendChild(pill);
    title.appendChild(left);
    title.appendChild(right);

    // Summary
    const summary = document.createElement('div');
    summary.className = 'mt-4 grid md:grid-cols-3 gap-3';

    const updated = document.createElement('div');
    updated.innerHTML = `<div class="text-xs text-slate-400">Last checked</div><div
class="font-medium">${record && record.lastUpdated ? record.lastUpdated : '—'}</div>`;

    const program = document.createElement('div');
    program.innerHTML = `<div class="text-xs text-slate-400">Program / Notes</div><div
class="font-medium">${record && record.details && record.details.program ?
escapeHtml(record.details.program) : '—'}</div>`;

    const bank = document.createElement('div');
    bank.innerHTML = `<div class="text-xs text-slate-400">Ledger</div><div
class="font-medium">${record && record.details && record.details.bank ?
escapeHtml(record.details.bank) : (record && record.details && record.details.bankOnFile ?
escapeHtml(record.details.bankOnFile) : '—')}</div>`;

    summary.appendChild(updated);
    summary.appendChild(program);
    summary.appendChild(bank);

    // Actionable instructions
    const actions = document.createElement('div');
    actions.className = 'mt-5';

    const heading = document.createElement('h3');
    heading.className = 'font-semibold';
    heading.textContent = 'Action & Fixes';
    actions.appendChild(heading);

    const list = document.createElement('div');
    list.className = 'mt-3 text-sm space-y-3 text-slate-700';

    if (!record) {
      list.innerHTML = `<div>We couldn't find any record for this identifier. Steps you can
try:</div>
```

```
        <ol class="list-decimal pl-5 mt-2 text-slate-600">
          <li>Check for typos (student ID / email).</li>
          <li>Try alternate email or enrollment number.</li>
          <li>Contact <strong>support@college.edu</strong> with your full name and admission
number.</li>
        </ol>`;
    } else if (status === 'seeded') {
      list.innerHTML = `<div>Your seeding looks good. Things you can do:</div>
        <ul class="list-disc pl-5 mt-2 text-slate-600">
          <li>Download the confirmation receipt from your student portal.</li>
          <li>If the credited amount doesn't match, contact accounts@college.edu with transaction
details.</li>
        </ul>`;
    } else if (status === 'pending') {
      list.innerHTML = `<div>The seeding is still pending.</div>
        <ol class="list-decimal pl-5 mt-2 text-slate-600">
          <li>Wait 24–48 hours for bank processing.</li>
          <li>If pending > 48 hours, raise a ticket at <strong>support@college.edu</strong> or your
student portal.</li>
        </ol>`;
    } else if (status === 'failed') {
      // Provide custom instructions based on error codes
      const d = record.details || {};
      if (d.code === 'MISSING_DOCS') {
        list.innerHTML = `<div>Failure reason: Missing documents.</div>
          <ul class="list-disc pl-5 mt-2 text-slate-600">
            <li>Missing: ${Array.isArray(d.missing) ? escapeHtml(d.missing.join(', ')) : 'documents'
}.</li>
            <li>Action: Upload the required documents to <strong>Student Portal →
Documents</strong> and re-check.</li>
            <li>If you uploaded already, wait 24 hours for verification or email
<strong>docs@college.edu</strong>.</li>
          </ul>`;
      } else if (d.code === 'BANK_MISMATCH') {
        list.innerHTML = `<div>Failure reason: Bank details mismatch.</div>
          <ol class="list-decimal pl-5 mt-2 text-slate-600">
            <li>Bank on file: <strong>${escapeHtml(d.bankOnFile || '—')}</strong></li>
            <li>Provided bank: <strong>${escapeHtml(d.providedBank || '—')}</strong></li>
            <li>Action: Update correct bank details at <strong>Profile → Bank Details</strong> or
contact <strong>accounts@college.edu</strong>.</li>
          </ol>`;
      } else {
        list.innerHTML = `<div>Seeding failed for an unknown reason.</div>
```

```
          <div class="mt-2 text-slate-600">Please contact <strong>accounts@college.edu</strong>
and include a screenshot of this check and your admission number.</div>`;
      }
    }

    actions.appendChild(list);

    // Quick links
    const quick = document.createElement('div');
    quick.className = 'mt-4 flex flex-wrap gap-2';
    quick.innerHTML = `
      <a href="#" onclick="alert('Open Student Portal (demo)')" class="px-3 py-2 rounded-md
text-sm bg-white border">Open Student Portal</a>
      <a href="#" onclick="alert('Open Upload Documents (demo)')" class="px-3 py-2 rounded-md
text-sm bg-white border">Upload Documents</a>
      <a href="#" onclick="copyReport('${escapeJs(JSON.stringify({identifier: query, record}))}')"
class="px-3 py-2 rounded-md text-sm bg-indigo-50 border">Copy Report</a>
    `;

    // Append all to result box
    resultBox.appendChild(title);
    resultBox.appendChild(summary);
    resultBox.appendChild(actions);
    resultBox.appendChild(quick);

    // Smooth scroll into view
    resultBox.scrollIntoView({ behavior: 'smooth', block: 'center' });
  }

  function escapeHtml(unsafe) {
    if (unsafe === undefined || unsafe === null) return '';
    return String(unsafe).replace(/[&<>"]+/g, function(match) {
      return ({'&':'&amp;','<':'&lt;','>':'&gt;','"':'&quot;'})[match];
    });
  }

  function escapeJs(s) {
    return s.replace(/'/g, "\\'").replace(/\n/g, '\\n');
  }

  function copyReport(jsonStr) {
    // Copy a short JSON report to clipboard
    navigator.clipboard.writeText(jsonStr).then(() => {
      alert('Report copied to clipboard (demo).');
    }).catch(() => alert('Unable to copy.'));
  }

  checkForm.addEventListener('submit', (e) => {
    e.preventDefault();
    const q = identifierInput.value.trim();
    if (!q) return;

    // Simulate a quick "instant" lookup latency with deterministic demo behavior
    const key = q.toLowerCase();

    // Show loading
    resultBox.classList.remove('hidden');
    resultBox.innerHTML = '<div class="flex items-center gap-3"><svg class="w-6 h-6 animate-spin" viewBox="0 0 24 24"><circle class="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" stroke-width="4" fill="none"></circle><path class="opacity-75" d="M4 12a8 8 0 018-8v8z" fill="currentColor"></path></svg><div>Checking status…</div></div>';

    setTimeout(() => {
      const record = sampleDatabase[key] || null;
      renderStatusCard(record, q);
    }, 450);
  });

  sampleBtn.addEventListener('click', () => {
    const keys = Object.keys(sampleDatabase);
    const pick = keys[Math.floor(Math.random()*keys.length)];
    identifierInput.value = pick;
    checkForm.dispatchEvent(new Event('submit', { cancelable: true }));
  });

  // Optional: prefill from URL like ?id=S2025001
  (function prefill() {
    const params = new URLSearchParams(location.search);
    if (params.has('id')) {
      identifierInput.value = params.get('id');
      checkForm.dispatchEvent(new Event('submit', { cancelable: true }));
    }
  })();

  </script></body>
</html>
```