```
import { useState, useEffect } from "react";

const SAMPLE_SOP = `Standard Operating Procedure: New Client Onboarding

Purpose: To standardize the process for bringing on new clients at our firm.

Step 1: Sales team sends a welcome email to client after contract is verbally agr
Step 2: Client fills out an intake form manually and emails it back as a PDF.
Step 3: Account manager reviews the intake form and manually inputs data into our
Step 4: Finance team creates an invoice manually in QuickBooks based on the propo
Step 5: Legal emails the contract for signature and waits for a reply.
Step 6: Once signed, account manager schedules kickoff call via email back-and-fo
Step 7: Project team is notified via group chat with a screenshot of the intake f
Step 8: Account manager manually creates a project folder in Google Drive and sha
Step 9: Client receives login credentials to our portal via a separate email.

Notes:
- End-to-end process currently takes 5 to 7 business days.
- Multiple departments are involved but there is no central coordination system.
- Clients frequently ask for status updates that we cannot quickly answer.
- Contracts sometimes sit unsigned for over a week with no follow-up mechanism.
- There is no formal checklist to confirm all steps are completed before kickoff.

const SYSTEM_PROMPT = `You are a senior operations analyst and business process i

{"bottlenecks": ["finding 1", "finding 2", "finding 3"], "risks": ["finding 1", "

Rules:
- Each array must contain exactly 3 to 5 findings
- Every finding must be a single, specific sentence tied to the actual content pr
- Never give generic advice — reference specific steps, people, or details from t
- Bottlenecks: where work slows down or creates waiting
- Risks: things that could go wrong, cause errors, or hurt quality / compliance
- Missing Steps: gaps in the process that should exist but don't
- Automation Opportunities: specific manual tasks that could be replaced or accel

const CATEGORIES = [
  { key: "bottlenecks", label: "Bottlenecks", icon: "⚡", color: "#ef4444", bg: "
  { key: "risks", label: "Risks", icon: "⚠️", color: "#f97316", bg: "rgba(249,115
  { key: "missing_steps", label: "Missing Steps", icon: "🔍", color: "#eab308", b
  { key: "automation_opportunities", label: "Automation Opportunities", icon: "🤖
];

const DOC_TYPES = ["SOP", "Workflow", "Meeting Notes", "Process Doc", "Other"];
```

```
export default function OperationsAuditAI() {
  const [docType, setDocType] = useState("SOP");
  const [content, setContent] = useState("");
  const [loading, setLoading] = useState(false);
  const [results, setResults] = useState(null);
  const [error, setError] = useState(null);
  const [copied, setCopied] = useState(false);

  useEffect(() => {
    const link = document.createElement("link");
    link.rel = "stylesheet";
    link.href = "https://fonts.googleapis.com/css2?family=Syne:wght@600;700;800&f
    document.head.appendChild(link);
    return () => { if (document.head.contains(link)) document.head.removeChild(li
  }, []);

  const analyze = async () => {
    if (!content.trim()) {
      setError("Please paste a document before analyzing.");
      return;
    }
    setLoading(true);
    setError(null);
    setResults(null);

    try {
      const response = await fetch("https://api.anthropic.com/v1/messages", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({
          model: "claude-sonnet-4-20250514",
          max_tokens: 1000,
          system: SYSTEM_PROMPT,
          messages: [{ role: "user", content: `Document Type: ${docType}\n\nDocum
        }),
      });

      const data = await response.json();
      const raw = data.content?.[0]?.text || "";
      const clean = raw.replace(/```json|```/g, "").trim();
      const parsed = JSON.parse(clean);
      setResults(parsed);
    } catch {
      setError("Analysis failed. Make sure the document has enough content and tr
    } finally {
      setLoading(false);
```

```
  }
};

const copyResults = () => {
  if (!results) return;
  const text = CATEGORIES.map(({ key, label }) => {
    const items = results[key] || [];
    return `${label}:\n${items.map((item, i) => `${i + 1}. ${item}`).join("\n")
  }).join("\n\n");
  navigator.clipboard.writeText(text).then(() => {
    setCopied(true);
    setTimeout(() => setCopied(false), 2000);
  });
};

const reset = () => { setResults(null); setContent(""); setError(null); };

return (
  <div style={{ fontFamily: "'DM Sans', sans-serif", background: "#16161a", min
    <style>{`
      @keyframes spin { to { transform: rotate(360deg); } }
      @keyframes fadeUp { from { opacity: 0; transform: translateY(16px); } to
      .audit-textarea:focus { border-color: rgba(184,155,114,0.5) !important; b
      .doc-btn:hover { border-color: rgba(184,155,114,0.4) !important; color: #
      .btn-primary:hover { transform: translateY(-2px); box-shadow: 0 8px 24px
      .btn-ghost:hover { border-color: rgba(255,255,255,0.25) !important; color
      .result-card { animation: fadeUp 0.4s ease both; }
      .result-card:nth-child(2) { animation-delay: 0.08s; }
      .result-card:nth-child(3) { animation-delay: 0.16s; }
      .result-card:nth-child(4) { animation-delay: 0.24s; }
    `}</style>

    {/* Header */}
    <div style={{ padding: "44px 44px 36px", borderBottom: "1px solid rgba(255,
      <div style={{ display: "inline-flex", alignItems: "center", gap: "7px", b
        ⚙️ AI-Powered Operations Tool
      </div>
      <h1 style={{ fontFamily: "'Syne', sans-serif", fontSize: "clamp(2.2rem, 5
        Operations Audit <span style={{ color: "#b89b72" }}>AI</span>
      </h1>
      <p style={{ color: "#9a9aa2", fontSize: "1rem", lineHeight: "1.65", maxWi
        Paste any SOP, workflow, or meeting notes. Get an instant audit identif
      </p>
    </div>

    <div style={{ padding: "40px 44px", maxWidth: "1100px" }}>
```

```
{/* Input panel */}
{!results && !loading && (
  <div style={{ animation: "fadeUp 0.5s ease both" }}>
    <div style={{ marginBottom: "24px" }}>
      <span style={{ display: "block", fontSize: "10px", fontWeight: "700
        Document Type
      </span>
      <div style={{ display: "flex", gap: "8px", flexWrap: "wrap" }}>
        {DOC_TYPES.map((t) => (
          <button
            key={t}
            className="doc-btn"
            onClick={() => setDocType(t)}
            style={{ padding: "8px 18px", borderRadius: "30px", fontSize:
          >
            {t}
          </button>
        ))}
      </div>
    </div>

    <span style={{ display: "block", fontSize: "10px", fontWeight: "700",
      Paste Your Document
    </span>
    <textarea
      className="audit-textarea"
      value={content}
      onChange={(e) => setContent(e.target.value)}
      placeholder={`Paste your ${docType} content here...`}
      style={{ width: "100%", minHeight: "240px", background: "rgba(255,2
    />

    <div style={{ display: "flex", gap: "12px", marginTop: "16px", flexWr
      <button
        className="btn-primary"
        onClick={analyze}
        style={{ padding: "13px 30px", borderRadius: "30px", background:
      >
        Analyze Document →
      </button>
      <button
        className="btn-ghost"
        onClick={() => { setContent(SAMPLE_SOP); setDocType("SOP"); }}
        style={{ padding: "13px 20px", borderRadius: "30px", background:
      >
        Load Sample SOP
      </button>
```

```
    </div>

    {error && (
      <div style={{ marginTop: "16px", padding: "12px 18px", background:
        {error}
      </div>
    )}
  </div>
)}

{/* Loading */}
{loading && (
  <div style={{ textAlign: "center", padding: "80px 40px" }}>
    <div style={{ width: "42px", height: "42px", border: "3px solid rgba(
    <p style={{ fontFamily: "'Syne', sans-serif", fontSize: "1.15rem", fo
    <p style={{ color: "#9a9aa2", fontSize: "14px" }}>Identifying bottlen
  </div>
)}

{/* Results */}
{results && (
  <>
    <div style={{ display: "flex", justifyContent: "space-between", align
      <div>
        <p style={{ fontSize: "11px", fontWeight: "600", letterSpacing: "
        <h2 style={{ fontFamily: "'Syne', sans-serif", fontSize: "1.5rem"
      </div>
      <div style={{ display: "flex", gap: "10px" }}>
        <button
          className="btn-ghost"
          onClick={copyResults}
          style={{ padding: "10px 18px", borderRadius: "30px", background
        >
          {copied ? "✓ Copied" : "Copy Results"}
        </button>
        <button
          className="btn-ghost"
          onClick={reset}
          style={{ padding: "10px 18px", borderRadius: "30px", background
        >
          New Audit
        </button>
      </div>
    </div>

    <div style={{ display: "grid", gridTemplateColumns: "repeat(auto-fit,
      {CATEGORIES.map(({ key, label, icon, color, bg, border }) => (
```

```
                <div
                  key={key}
                  className="result-card"
                  style={{ background: bg, border: `1px solid ${border}`, borderL
                >
                  <div style={{ display: "flex", alignItems: "center", gap: "10px
                    <span style={{ fontSize: "18px" }}>{icon}</span>
                    <span style={{ fontFamily: "'Syne', sans-serif", fontSize: "1
                  </div>
                  {(results[key] || []).map((finding, i) => (
                    <div key={i} style={{ display: "flex", gap: "10px", marginBot
                      <span style={{ fontFamily: "'JetBrains Mono', monospace", f
                        {String(i + 1).padStart(2, "0")}
                      </span>
                      <span style={{ fontSize: "13px", color: "#cfcfd4", lineHeig
                    </div>
                  ))}
                </div>
              ))}
            </div>

            <div style={{ marginTop: "32px", padding: "20px 24px", background: "r
              <p style={{ fontSize: "13px", color: "#9a9aa2", margin: 0 }}>
                Built by <span style={{ color: "#d8c6a6", fontWeight: "600" }}>Lo
              </p>
              <p style={{ fontSize: "12px", color: "#6b6b73", margin: 0, fontFami
                {Object.values(results).flat().length} findings across 4 categori
              </p>
            </div>
          </>
        )}
      </div>
    </div>
  );
}
```