<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>XzeCure ABG Tool</title>  
 <script src="https://cdn.tailwindcss.com"></script>  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.29.1/moment.min.js"></script>  
 <style>  
 @import url('https://fonts.googleapis.com/css2?family=Inter:wght@400;600;800&display=swap');  
 body {  
 font-family: 'Inter', sans-serif;  
 background-color: #f1f5f9; /\* Slate 100 \*/  
 }  
 .card-shadow {  
 box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1), 0 4px 6px -4px rgba(0, 0, 0, 0.05);  
 }  
 input[type="number"]::-webkit-inner-spin-button,  
 input[type="number"]::-webkit-outer-spin-button {  
 -webkit-appearance: none;  
 margin: 0;  
 }  
 input[type="number"] {  
 -moz-appearance: textfield;  
 }  
 </style>  
</head>  
<body class="p-4 sm:p-6 md:p-8 flex flex-col min-h-screen">  
  
 <!-- Header and Emergency Button -->  
 <header class="mb-6 bg-white p-4 rounded-xl card-shadow flex flex-col md:flex-row justify-between items-center">  
 <div>  
 <h1 class="text-3xl font-extrabold text-blue-900">XzeCure ABG Tool</h1>  
 <p class="text-sm text-gray-600">Arterial Blood Gas Analysis & Metabolic Status Interpreter</p>  
 </div>  
 <a href="https://api.whatsapp.com/send?phone=918200095781&text=My%20Patient%20needs%20urgent%20ICU%20care.%20Where%20can%20we%20refer%3F"   
 target="\_blank"  
 class="mt-4 md:mt-0 px-6 py-3 bg-red-600 text-white font-bold rounded-lg hover:bg-red-700 transition duration-150 shadow-lg text-center whitespace-nowrap">  
 🚨 Emergency (Refer to ICU)  
 </a>  
 </header>  
  
 <!-- Main ABG Content Area -->  
 <main class="flex-grow bg-white p-4 sm:p-6 rounded-xl card-shadow mb-8">  
 <h2 class="text-xl font-semibold mb-4 text-blue-700 border-b pb-2">ABG Interpretation and Metabolic Status</h2>  
   
 <!-- ABG Inputs -->  
 <div id="abg-inputs" class="grid grid-cols-2 sm:grid-cols-3 lg:grid-cols-4 gap-3 mb-6">  
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">pH <span class="text-red-500">\*</span></label><input type="number" id="input\_pH" value="7.18" step="0.01" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">PaCO₂ (mmHg) <span class="text-red-500">\*</span></label><input type="number" id="input\_paco2" value="28" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">HCO₃⁻ (mmol/L) <span class="text-red-500">\*</span></label><input type="number" id="input\_hco3" value="8" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
   
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">Na (mmol/L)</label><input type="number" id="input\_na" value="135" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">Cl (mmol/L)</label><input type="number" id="input\_cl" value="95" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">K (mmol/L)</label><input type="number" id="input\_k" value="4.5" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
   
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">PaO₂ (mmHg)</label><input type="number" id="input\_pao2" value="90" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">FiO₂ (0.21 - 1.0)</label><input type="number" id="input\_fio2" value="0.21" step="0.01" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">Age (Years)</label><input type="number" id="input\_age" value="30" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">Albumin (g/dL)</label><input type="number" id="input\_albumin" value="4.0" step="0.1" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
 <div class="col-span-1"><label class="block text-xs font-medium text-gray-700">Lactate (mmol/L)</label><input type="number" id="input\_lactate" value="3.2" step="0.1" class="w-full p-2 border border-gray-300 rounded-md text-sm"></div>  
 </div>  
  
 <button onclick="interpretABG()" class="w-full py-3 bg-blue-600 text-white font-bold rounded-lg hover:bg-blue-700 transition duration-150 mb-4 shadow-md">  
 Interpret ABG & Metabolic Status  
 </button>  
  
 <!-- Loading Indicator & Error Message -->  
 <div id="abg-loading" class="hidden text-center text-blue-500 font-medium my-4">Analyzing complex acid-base interactions...</div>  
 <div id="abg-error" class="text-red-600 text-sm hidden p-3 bg-red-50 border border-red-200 rounded-md mb-4"></div>  
  
 <!-- Results Output -->  
 <div id="abg-output" class="mt-6 border-t pt-4 hidden">  
 <div id="result-summary"></div>  
 <button id="toggle-log" onclick="toggleCalculationLog()" class="mt-3 text-sm font-medium text-blue-600 hover:text-blue-800 hidden">Show Calculation Log</button>  
 <div id="calculation-log" class="mt-3 p-3 bg-gray-100 rounded-lg text-xs hidden whitespace-pre-wrap overflow-x-auto"></div>  
 </div>  
 </main>  
  
 <!-- Promotional Footer -->  
 <footer class="mt-8 pt-6 border-t border-gray-300 bg-white p-6 rounded-xl card-shadow">  
 <h3 class="text-lg font-bold text-gray-800 mb-4">📚 Learn Critical Care with Dr. XzeCure</h3>  
   
 <div class="flex flex-col md:flex-row md:space-x-8 space-y-6 md:space-y-0 text-sm">  
   
 <!-- Courses and Social -->  
 <div class="md:w-1/3">  
 <h4 class="font-semibold text-blue-700 mb-2 border-b pb-1">Courses & Social Media</h4>  
 <p class="mb-2"><a href="https://wa.link/ex9dxa" target="\_blank" class="text-green-600 font-bold hover:underline">Enroll for ICU course, Starting 20/10/25</a></p>  
 <p class="mb-2">Follow for more: <a href="https://www.instagram.com/askdr.xze" target="\_blank" class="text-pink-600 hover:underline font-medium">IG: @askdr.xze</a></p>  
 <p class="mb-2">All Links: <a href="https://linktr.ee/xzecure" target="\_blank" class="text-gray-600 hover:underline">https://linktr.ee/xzecure</a></p>  
 </div>  
  
 <!-- Books & E-books -->  
 <div class="md:w-2/3 grid grid-cols-1 sm:grid-cols-3 gap-6">  
   
 <!-- Book 1 -->  
 <div>  
 <h4 class="font-semibold text-blue-700 mb-2 border-b pb-1">📘 1. ICU ESSENTIALS</h4>  
 <ul class="space-y-1">  
 <li>Paperback: <a href="https://store.pothi.com/book/dr-kenil-shah-icu-essentials-50-common-case-presentations/" target="\_blank" class="text-blue-500 hover:underline">Pothi Store</a></li>  
 <li>Audio: <a href="https://play.google.com/store/audiobooks/details?id=AQAAAEAqIFhgYM" target="\_blank" class="text-orange-500 hover:underline">Google Audiobooks</a></li>  
 <li>E-book: <a href="https://amzn.in/d/95JsyRX" target="\_blank" class="text-blue-500 hover:underline">Amazon</a> | <a href="https://play.google.com/store/books/details?id=FldpEQAAQBAJ" target="\_blank" class="text-blue-500 hover:underline">Google Books</a></li>  
 </ul>  
 </div>  
  
 <!-- Book 2 -->  
 <div>  
 <h4 class="font-semibold text-blue-700 mb-2 border-b pb-1">📕 2. Ultimate Interpretation Handbook</h4>  
 <ul class="space-y-1">  
 <li>Pothi: <a href="https://store.pothi.com/book/ebook-dr-kenil-shah-ultimate-interpretation-handbook/" target="\_blank" class="text-blue-500 hover:underline">Pothi Store</a></li>  
 <li>E-book: <a href="https://kdp.amazon.com/amazon-dp-action/in/dualbookshelf.marketplacelink/B0FG51YQ52" target="\_blank" class="text-blue-500 hover:underline">Amazon</a> | <a href="https://play.google.com/store/books/details?id=7mpsEQAAQBAJ" target="\_blank" class="text-blue-500 hover:underline">Google Books</a></li>  
 </ul>  
 </div>  
  
 <!-- Book 3 -->  
 <div>  
 <h4 class="font-semibold text-blue-700 mb-2 border-b pb-1">📗 3. Say It Simple</h4>  
 <ul class="space-y-1">  
 <li>Pothi: <a href="https://store.pothi.com/book/ebook-dr-kenil-shah-ultimate-interpretation-handbook/" target="\_blank" class="text-blue-500 hover:underline">Pothi Store</a></li>  
 <li>E-book: <a href="https://kdp.amazon.com/amazon-dp-action/in/dualbookshelf.marketplacelink/B0FFMSJZP2" target="\_blank" class="text-blue-500 hover:underline">Amazon</a> | <a href="https://play.google.com/store/books/details?id=unRpEQAAQBAJ" target="\_blank" class="class="text-blue-500 hover:underline">Google Books</a></li>  
 </ul>  
 </div>  
 </div>  
 </div>  
 </footer>  
  
  
 <!-- Confirmation Modal (for Management Suggestions) -->  
 <div id="confirmation-modal" class="fixed inset-0 bg-black bg-opacity-60 flex items-center justify-center z-50 hidden p-4">  
 <div class="bg-white p-6 rounded-xl w-full max-w-lg card-shadow">  
 <h3 class="text-xl font-bold text-red-700 mb-4 border-b pb-2">🚨 Confirm Clinical Action</h3>  
 <p id="modal-message" class="mb-4 text-gray-700"></p>  
 <p class="text-sm italic text-gray-500 mb-6">Final responsibility lies with the treating clinician. This action will be logged for audit purposes.</p>  
 <div class="flex justify-end space-x-3">  
 <button onclick="document.getElementById('confirmation-modal').classList.add('hidden')" class="px-4 py-2 text-gray-700 bg-gray-200 rounded-lg hover:bg-gray-300 transition">  
 Cancel  
 </button>  
 <button onclick="confirmAction()" class="px-4 py-2 bg-red-600 text-white rounded-lg hover:bg-red-700 transition">  
 Confirm & Log  
 </button>  
 </div>  
 </div>  
 </div>  
  
  
 <script>  
 // Global Constants  
 const ALGORITHM\_VERSION = '3.0.0-ABG-Only';  
 const NORMAL\_AG = 12;  
 const NORMAL\_HCO3 = 24;  
 const NORMAL\_PCO2 = 40;  
 const NORMAL\_ALBUMIN\_GDL = 4.0;  
 const PH\_WATER = 47;   
 const RESPIRATORY\_QUOTIENT = 0.8;   
  
 // --- Utility Functions ---  
  
 /\*\* Returns an ISO 8601 timestamp string. \*/  
 function now\_iso() {  
 return moment().toISOString();  
 }  
  
 // --- ABG Interpretation Logic (Identical to previous perfect version) ---  
  
 function parseABGInputs() {  
 const getVal = (id) => {  
 const el = document.getElementById(id);  
 const val = parseFloat(el.value);  
 return isNaN(val) ? null : val;  
 };  
  
 const inputs = {  
 pH: getVal('input\_pH'), paco2: getVal('input\_paco2'), hco3: getVal('input\_hco3'),  
 pao2: getVal('input\_pao2'), fio2: getVal('input\_fio2'), na: getVal('input\_na'),  
 k: getVal('input\_k'), cl: getVal('input\_cl'), albumin: getVal('input\_albumin'),  
 lactate: getVal('input\_lactate'), age: getVal('input\_age'), pb: 760,  
 };  
  
 if (inputs.pH === null || inputs.paco2 === null || inputs.hco3 === null) {  
 return { error: "pH, PaCO₂, and HCO₃⁻ are required inputs." };  
 }  
 if (inputs.pH < 6.8 || inputs.pH > 8.0) {  
 return { error: `pH value (${inputs.pH}) is physiologically implausible.` };  
 }  
 if (inputs.paco2 < 5 || inputs.paco2 > 200) {  
 return { error: `PaCO₂ value (${inputs.paco2}) is physiologically implausible.` };  
 }  
 return inputs;  
 }  
  
 function interpretABGAlgorithm(input) {  
 const { pH, paco2, hco3, pao2, fio2, na, cl, albumin, lactate, age, pb } = input;  
 const calc\_log = [];  
 let mixed = false;  
 let compensation = {};  
 let ag = null;  
 let corrected\_ag = null;  
 let delta\_ratio = null;  
 let a\_a = null;  
 let pf = null;  
  
 let acid\_base\_state = "near-normal";  
 if (pH < 7.35) acid\_base\_state = "acidosis";  
 else if (pH > 7.45) acid\_base\_state = "alkalosis";  
 calc\_log.push(`pH -> ${pH.toFixed(2)} => ${acid\_base\_state}`);  
  
 const paCO2\_dir = paco2 > 45 ? "high" : (paco2 < 35 ? "low" : "normal");  
 const HCO3\_dir = hco3 > 28 ? "high" : (hco3 < 22 ? "low" : "normal");  
 calc\_log.push(`PaCO2 ${paco2} (${paCO2\_dir}), HCO3 ${hco3} (${HCO3\_dir})`);  
  
 let primary\_candidates = [];  
 if (acid\_base\_state === "acidosis") {  
 if (paCO2\_dir === "high") primary\_candidates.push("Respiratory Acidosis");  
 if (HCO3\_dir === "low") primary\_candidates.push("Metabolic Acidosis");  
 } else if (acid\_base\_state === "alkalosis") {  
 if (paCO2\_dir === "low") primary\_candidates.push("Respiratory Alkalosis");  
 if (HCO3\_dir === "high") primary\_candidates.push("Metabolic Alkalosis");  
 }  
   
 let ag\_check\_value = null;  
 if (na !== null && cl !== null) {  
 ag = na - (cl + hco3);  
 ag\_check\_value = ag;  
  
 if (albumin !== null) {  
 corrected\_ag = ag + (2.5 \* (NORMAL\_ALBUMIN\_GDL - albumin));  
 ag\_check\_value = corrected\_ag;  
 }  
  
 if (ag\_check\_value > NORMAL\_AG) {  
 if (primary\_candidates.includes("Metabolic Acidosis") === false && HCO3\_dir === "low") {  
 primary\_candidates.push("High AG Metabolic Acidosis");  
 }  
 const deltaAG = ag\_check\_value - NORMAL\_AG;  
 const deltaHCO3 = NORMAL\_HCO3 - hco3;  
   
 if (deltaHCO3 !== 0) {  
 delta\_ratio = parseFloat((deltaAG / deltaHCO3).toFixed(2));  
 }  
 }  
 }  
  
 const tolerance = 2.0;  
 if (primary\_candidates.includes("Metabolic Acidosis")) {  
 const expected\_paco2 = 1.5 \* hco3 + 8;  
 let status;  
 if (Math.abs(paco2 - expected\_paco2) <= tolerance) status = "appropriate";  
 else if (paco2 > expected\_paco2 + tolerance) { status = "inappropriate - concurrent Resp Acidosis"; mixed = true; }  
 else { status = "inappropriate - concurrent Resp Alkalosis"; mixed = true; }  
 compensation.metabolic\_acidosis\_winter = { expected\_paco2: parseFloat(expected\_paco2.toFixed(1)), observed\_vs\_expected: status };  
 }  
   
 if (primary\_candidates.includes("Respiratory Acidosis")) {  
 const delta\_pco2 = paco2 - NORMAL\_PCO2;  
 const expected\_hco3\_acute = NORMAL\_HCO3 + (1 \* (delta\_pco2 / 10));  
 const expected\_hco3\_chronic = NORMAL\_HCO3 + (3.5 \* (delta\_pco2 / 10));  
 let status;  
 const diff\_acute = Math.abs(hco3 - expected\_hco3\_acute);  
 const diff\_chronic = Math.abs(hco3 - expected\_hco3\_chronic);  
 if (diff\_acute < 3) status = "appropriate - acute";  
 else if (diff\_chronic < 3) status = "appropriate - chronic";  
 else { status = "inappropriate - consider mixed"; mixed = true; }  
 compensation.respiratory\_acidosis = { expected\_hco3\_acute: parseFloat(expected\_hco3\_acute.toFixed(1)), expected\_hco3\_chronic: parseFloat(expected\_hco3\_chronic.toFixed(1)), observed\_vs\_expected: status };  
 }  
  
 if (acid\_base\_state === "near-normal" && primary\_candidates.length === 0) {  
 if (paco2\_dir === "high" && HCO3\_dir === "high") {  
 primary\_candidates.push("Chronic Respiratory Acidosis (Fully Compensated)");  
 } else if (paco2\_dir === "low" && HCO3\_dir === "low") {  
 primary\_candidates.push("Chronic Respiratory Alkalosis (Fully Compensated)");  
 } else {  
 primary\_candidates.push("Fully Compensated Disorder (Requires context)");  
 }  
 }  
  
 let aa\_interpret = null;  
 let pf\_interpret = null;  
 let normal\_Aa = null;  
  
 if (pao2 !== null && fio2 !== null && fio2 >= 0.21 && fio2 <= 1.0) {  
 const PAO2 = fio2 \* (pb - PH\_WATER) - (paco2 / RESPIRATORY\_QUOTIENT);  
 const A\_a\_value = PAO2 - pao2;  
 a\_a = { value: parseFloat(A\_a\_value.toFixed(1)) };  
   
 normal\_Aa = age !== null ? (age / 4.0) + 4.0 : 10.0;  
 a\_a.normal\_for\_age = parseFloat(normal\_Aa.toFixed(1));  
  
 if (A\_a\_value <= normal\_Aa) { aa\_interpret = "A-a gradient is normal (Hypoventilation or normal gas exchange)."; }   
 else { aa\_interpret = "A-a gradient is elevated (V/Q mismatch or shunt likely)."; }  
 a\_a.interpretation = aa\_interpret;  
  
 const PF\_VALUE = pao2 / fio2;  
 pf = { value: parseFloat(PF\_VALUE.toFixed(1)) };  
   
 if (PF\_VALUE < 100) pf\_interpret = "Severe hypoxemia (ARDS criteria severe)";  
 else if (PF\_VALUE < 200) pf\_interpret = "Moderate hypoxemia (ARDS criteria moderate)";  
 else if (PF\_VALUE < 300) pf\_interpret = "Mild hypoxemia (ARDS criteria mild)";  
 else pf\_interpret = "Normal oxygenation";  
 pf.interpretation = pf\_interpret;  
 }  
   
 const mgmt = [];  
 if (pf && pf.value < 200) { mgmt.push({ urgency: 'immediate', action: `Optimize oxygenation: PF ratio (${pf.value}) suggests Moderate-Severe Hypoxemia. Increase FiO2/PEEP, consider Prone Positioning.` }); }  
 if (paco2 > 60 && pH < 7.25 && primary\_candidates.includes("Respiratory Acidosis")) { mgmt.push({ urgency: 'immediate', action: 'Severe Respiratory Acidosis: Increase minute ventilation or prepare for intubation.' }); }  
 if (ag\_check\_value > NORMAL\_AG && ag\_check\_value !== null) { mgmt.push({ urgency: 'urgent', action: 'Investigate High AG Metabolic Acidosis: Check ketones (DKA), renal function, toxicology.' }); }  
 if (lactate !== null && lactate > 4.0) { mgmt.push({ urgency: 'immediate', action: `Severe Lactic Acidosis (Lactate ${lactate}): Initiate immediate volume resuscitation and address source of shock.` }); }  
 mgmt.push({ urgency: 'follow-up', action: 'Repeat ABG after critical interventions in 30–60 minutes or sooner if unstable.' });  
   
 let max\_confidence = 0.6;  
 const final\_diagnosis\_list = primary\_candidates.map(cand => {  
 const ph\_dev = Math.abs(7.4 - pH);  
 let conf = Math.min(0.95, 0.4 + ph\_dev \* 0.8);  
 max\_confidence = Math.max(max\_confidence, conf);  
 return { label: cand, confidence: parseFloat(conf.toFixed(2)), rationale: `pH ${acid\_base\_state}, PaCO2 ${paCO2\_dir}, HCO3 ${HCO3\_dir}.` };  
 });  
  
 if (mixed) {  
 final\_diagnosis\_list.push({ label: 'Mixed Acid-Base Disorder Confirmed', confidence: 0.85, rationale: 'Compensation limits exceeded or conflicting primary changes detected.' });  
 }  
  
 const confidence = Math.max(0.1, Math.min(0.99, max\_confidence));  
  
 return {  
 primary\_diagnoses: final\_diagnosis\_list,  
 mixed\_disorder\_detected: mixed,  
 compensation\_status: compensation,  
 anion\_gap: { value: ag, corrected: corrected\_ag },  
 delta\_ratio: delta\_ratio,  
 a\_a\_gradient: a\_a,  
 pf\_ratio: pf,  
 management\_suggestions: mgmt,  
 calculation\_log: calc\_log,  
 confidence\_score: parseFloat(confidence.toFixed(2)),  
 timestamp: now\_iso(),  
 algorithm\_version: ALGORITHM\_VERSION  
 };  
 }  
  
 async function interpretABG() {  
 document.getElementById('abg-loading').classList.remove('hidden');  
 document.getElementById('abg-output').classList.add('hidden');  
 document.getElementById('abg-error').classList.add('hidden');  
 document.getElementById('toggle-log').classList.add('hidden');  
 document.getElementById('calculation-log').classList.add('hidden');  
  
 const inputs = parseABGInputs();  
 if (inputs.error) {  
 document.getElementById('abg-error').innerText = inputs.error;  
 document.getElementById('abg-error').classList.remove('hidden');  
 document.getElementById('abg-loading').classList.add('hidden');  
 return;  
 }  
  
 try {  
 const result = interpretABGAlgorithm(inputs);  
 const summaryDiv = document.getElementById('result-summary');  
 const primaryDiag = result.primary\_diagnoses[0];  
  
 let html = `  
 <div class="space-y-4">  
 <div class="p-4 rounded-xl ${result.mixed\_disorder\_detected || result.pH < 7.35 || result.pH > 7.45 ? 'bg-red-100 border border-red-300' : 'bg-green-100 border border-green-300'}">  
 <h3 class="font-bold text-xl ${result.mixed\_disorder\_detected ? 'text-red-700' : 'text-blue-700'}">  
 Primary Finding: ${primaryDiag.label}  
 <span class="text-sm font-normal text-gray-600">(Conf: ${result.confidence\_score})</span>  
 </h3>  
 ${result.primary\_diagnoses.slice(1).map(d => `<p class="text-sm italic text-red-600 ml-1">Mixed/Secondary: ${d.label}</p>`).join('')}  
 </div>  
   
 <h4 class="font-semibold text-gray-700 mt-4">Key Metrics:</h4>  
 <div class="grid grid-cols-2 md:grid-cols-4 gap-3 text-sm p-3 border rounded-lg bg-gray-50">  
 <div><span class="font-medium text-gray-600">pH / State:</span> <span class="font-bold">${inputs.pH}</span></div>  
 <div><span class="font-medium text-gray-600">PaCO₂:</span> <span class="font-bold">${inputs.paco2} mmHg</span></div>  
 <div><span class="font-medium text-gray-600">HCO₃⁻:</span> <span class="font-bold">${inputs.hco3} mmol/L</span></div>  
 <div><span class="font-medium text-gray-600">AG (Corrected):</span> <span class="font-bold">${result.anion\_gap.corrected !== null ? result.anion\_gap.corrected.toFixed(1) : (result.anion\_gap.value !== null ? result.anion\_gap.value.toFixed(1) : 'N/A')}</span></div>  
 ${result.delta\_ratio ? `<div><span class="font-medium text-gray-600">Delta Ratio:</span> <span class="font-bold text-orange-600">${result.delta\_ratio}</span></div>` : ''}  
 ${result.pf\_ratio ? `<div><span class="font-medium text-gray-600">PF Ratio:</span> <span class="font-bold ${result.pf\_ratio.value < 300 ? 'text-red-600' : 'text-green-600'}">${result.pf\_ratio.value}</span></div>` : ''}  
 ${result.a\_a\_gradient ? `<div><span class="font-medium text-gray-600">A-a Grad:</span> <span class="font-bold">${result.a\_a\_gradient.value} mmHg</span></div>` : ''}  
 ${inputs.lactate ? `<div><span class="font-medium text-gray-600">Lactate:</span> <span class="font-bold ${inputs.lactate > 4.0 ? 'text-red-700' : (inputs.lactate > 2.0 ? 'text-orange-600' : 'text-green-600')}">${inputs.lactate} mmol/L</span></div>` : ''}  
 </div>  
  
 <h4 class="font-semibold text-gray-700 pt-3">Suggested Management Actions:</h4>  
 <ul class="list-disc ml-5 text-sm space-y-1">  
 ${result.management\_suggestions.map(m =>   
 `<li class="${m.urgency === 'immediate' ? 'text-red-600 font-bold' : m.urgency === 'urgent' ? 'text-orange-600' : 'text-gray-700'}">  
 <span class="font-bold uppercase text-xs">${m.urgency}:</span> ${m.action}  
 <button onclick="showConfirmationModal('${m.action}')" class="ml-2 px-2 py-0.5 text-xs bg-red-100 text-red-600 rounded-full hover:bg-red-200 shadow-sm">Confirm</button>  
 </li>`).join('')}  
 </ul>  
 </div>  
 `;  
   
 summaryDiv.innerHTML = html;  
 document.getElementById('calculation-log').innerText = result.calculation\_log.join('\n');  
 document.getElementById('toggle-log').classList.remove('hidden');  
  
 } catch (e) {  
 console.error("ABG Interpretation Error:", e);  
 document.getElementById('abg-error').innerText = "An unexpected error occurred during calculation. Please check inputs.";  
 document.getElementById('abg-error').classList.remove('hidden');  
 } finally {  
 document.getElementById('abg-loading').classList.add('hidden');  
 document.getElementById('abg-output').classList.remove('hidden');  
 }  
 }  
  
 function toggleCalculationLog() {  
 const log = document.getElementById('calculation-log');  
 const button = document.getElementById('toggle-log');  
 if (log.classList.contains('hidden')) {  
 log.classList.remove('hidden');  
 button.innerText = 'Hide Calculation Log';  
 } else {  
 log.classList.add('hidden');  
 button.innerText = 'Show Calculation Log';  
 }  
 }  
  
 function showConfirmationModal(action) {  
 document.getElementById('modal-message').innerHTML = `You are confirming the following action for audit: <br><br><strong class="text-lg">${action}</strong>`;  
 document.getElementById('confirmation-modal').classList.remove('hidden');  
 window.currentAction = action;   
 }  
  
 function confirmAction() {  
 console.log(`[AUDIT LOG - ${now\_iso()}] User confirmed action: ${window.currentAction}`);  
 document.getElementById('confirmation-modal').classList.add('hidden');  
 alert('Action confirmed and logged for audit (check console).');  
 }  
   
 // Initial setup on load  
 window.onload = function() {  
 interpretABG();   
 };  
  
 </script>  
</body>  
</html>