<!DOCTYPE html>

<html>

<head>

<base target="\_top">

<script src="https://cdn.tailwindcss.com"></script>

<style>

body, html {

background-color: #111827;

color: #f3f4f6;

font-family: sans-serif;

padding: 16px;

}

.icon {

width: 1.25rem;

height: 1.25rem;

display: inline-block;

vertical-align: middle;

}

pre {

white-space: pre-wrap;

word-wrap: break-word;

}

.loader {

border: 4px solid #f3f3f3;

border-top: 4px solid #4f46e5;

border-radius: 50%;

width: 24px;

height: 24px;

animation: spin 1s linear infinite;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

</style>

</head>

<body class="bg-gray-900 text-gray-100">

<div class="space-y-6">

<header class="text-center">

<h1 class="text-2xl font-bold text-white">G-Assistant Code Reviewer</h1>

<p class="text-sm text-gray-400">Powered by Gemini</p>

</header>

<div>

<label for="code-input" class="block text-sm font-medium text-gray-300 mb-2">

Paste Apps Script Code Here

</label>

<textarea id="code-input"

placeholder="// Paste your .gs code here..."

class="w-full h-80 p-3 font-mono text-sm bg-gray-800 border border-gray-700 rounded-lg focus:ring-2 focus:ring-indigo-500 focus:border-indigo-500 resize-y"

spellcheck="false"></textarea>

</div>

<div>

<button id="review-button"

class="w-full flex items-center justify-center gap-2 px-4 py-2.5 border border-transparent text-base font-medium rounded-md shadow-sm text-white bg-indigo-600 hover:bg-indigo-700 focus:ring-2 focus:ring-indigo-500 disabled:bg-indigo-900/50 disabled:cursor-not-allowed transition-all duration-200 group">

<span id="button-icon">

<svg xmlns="http://www.w3.org/2000/svg"

viewBox="0 0 20 20"

fill="currentColor"

class="icon text-indigo-300 group-hover:scale-110 transition-transform">

<path fill-rule="evenodd"

d="M10 2a.75.75 0 01.75.75v.263a3.5 3.5 0 014.288 4.288H15.25a.75.75 0 010 1.5h-.263a3.5 3.5 0 01-4.288 4.288v.263a.75.75 0 01-1.5 0v-.263a3.5 3.5 0 01-4.288-4.288H4.75a.75.75 0 010-1.5h.263A3.5 3.5 0 019.25 4.75v-.263A.75.75 0 0110 2z"/>

</svg>

</span>

<span id="button-text">Review Code</span>

</button>

</div>

<div id="error-container" class="hidden p-4 bg-red-900/50 border border-red-700 text-red-200 rounded-lg"></div>

<div id="output-container" class="mt-6 space-y-4"></div>

</div>

<script>

const reviewButton = document.getElementById('review-button');

const buttonText = document.getElementById('button-text');

const buttonIcon = document.getElementById('button-icon');

const codeInput = document.getElementById('code-input');

const outputContainer = document.getElementById('output-container');

const errorContainer = document.getElementById('error-container');

const originalButtonIconHTML = buttonIcon.innerHTML;

reviewButton.addEventListener('click', () => {

const code = codeInput.value;

if (!code.trim()) return showError("Please enter some code to review.");

setLoading(true);

google.script.run

.withSuccessHandler(displayReview)

.withFailureHandler(showError)

.reviewCodeForSidebar(code);

});

function setLoading(isLoading) {

reviewButton.disabled = isLoading;

buttonText.textContent = isLoading ? 'Reviewing...' : 'Review Code';

buttonIcon.innerHTML = isLoading ? '<div class="loader"></div>' : originalButtonIconHTML;

outputContainer.innerHTML = isLoading ? '<p class="text-center text-gray-400">Analyzing your code...</p>' : '';

errorContainer.classList.add('hidden');

}

function showError(error) {

setLoading(false);

const msg = error?.message || String(error);

errorContainer.textContent = `Error: ${msg}`;

errorContainer.classList.remove('hidden');

outputContainer.innerHTML = '';

}

function displayReview(review) {

setLoading(false);

if (typeof review !== 'object') return showError("Invalid review object received.");

const scoreColor = review.overallScore < 50 ? 'text-red-400' : (review.overallScore < 80 ? 'text-yellow-400' : 'text-green-400');

let html = `

<div class="p-4 bg-gray-800 rounded-xl border border-gray-700">

<h2 class="text-xl font-bold text-gray-100 mb-2">Review Summary</h2>

<div class="flex items-center gap-4">

<div class="text-3xl font-bold ${scoreColor}">${review.overallScore}</div>

<p class="text-gray-300 flex-1">${escapeHtml(review.summary)}</p>

</div>

</div>

`;

if (Array.isArray(review.feedbackItems) && review.feedbackItems.length > 0) {

review.feedbackItems.forEach(item => {

html += createFeedbackCard(item);

});

} else {

html += '<div class="text-center py-8 text-gray-400">No specific feedback was generated. Your code looks great!</div>';

}

outputContainer.innerHTML = html;

}

function getFeedbackTypeStyles(type) {

const styles = {

BUG: { title: 'Bug Detected', bg: 'bg-red-900/30', border: 'border-red-700/50', text: 'text-red-300' },

SUGGESTION: { title: 'Suggestion', bg: 'bg-blue-900/30', border: 'border-blue-700/50', text: 'text-blue-300' },

STYLE: { title: 'Style Tip', bg: 'bg-purple-900/30', border: 'border-purple-700/50', text: 'text-purple-300' },

PRAISE: { title: 'Good Practice', bg: 'bg-green-900/30', border: 'border-green-700/50', text: 'text-green-300' }

};

return styles[type] || { title: 'Feedback', bg: 'bg-gray-700', border: 'border-gray-600', text: 'text-gray-300' };

}

function createFeedbackCard(item) {

const s = getFeedbackTypeStyles(item.type);

const line = item.line ? `<span class="ml-auto text-xs font-mono bg-gray-700 text-gray-300 px-2 py-1 rounded">Line: ${item.line}</span>` : '';

const suggestion = item.suggestion ? `

<div class="mt-3 pt-3 border-t border-gray-700/50">

<p class="text-sm font-semibold text-gray-400 mb-1">Suggestion:</p>

<pre class="p-3 bg-gray-900/70 rounded-md text-sm font-mono text-gray-200 overflow-x-auto"><code>${escapeHtml(item.suggestion)}</code></pre>

</div>` : '';

return `

<div class="p-4 rounded-lg border ${s.bg} ${s.border} shadow-md">

<div class="flex items-center gap-3">

<h3 class="text-lg font-semibold ${s.text}">${s.title}</h3>

${line}

</div>

<p class="mt-3 text-gray-300">${escapeHtml(item.description)}</p>

${suggestion}

</div>

`;

}

function escapeHtml(text) {

return typeof text !== 'string' ? '' :

text.replace(/&/g, "&amp;")

.replace(/</g, "&lt;")

.replace(/>/g, "&gt;