<!DOCTYPE html>

<html lang="ar" dir="rtl">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>مساعد Apps Script الذكي</title>

<style>

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

direction: rtl;

padding: 0;

margin: 0;

background-color: #1a1a1a; /\* خلفية داكنة جداً \*/

color: #e0e0e0; /\* لون نص فاتح \*/

}

.container {

padding: 20px;

}

h2 {

text-align: center;

color: #7bbaff; /\* أزرق فاتح للعناوين \*/

margin-bottom: 25px;

font-size: 1.8em;

}

.tabs {

display: flex;

justify-content: center;

margin-bottom: 20px;

border-bottom: 1px solid #333; /\* حدود داكنة \*/

}

.tab-button {

background-color: #2c2c2c; /\* خلفية أزرار التبويب غير النشطة \*/

border: none;

padding: 10px 15px;

cursor: pointer;

font-size: 1em;

font-weight: bold;

color: #b0b0b0; /\* لون نص أزرار التبويب غير النشطة \*/

border-radius: 5px 5px 0 0;

transition: background-color 0.3s ease, color 0.3s ease;

}

.tab-button.active {

background-color: #007bff; /\* أزرق ساطع للتبويب النشط \*/

color: white;

border-bottom: 2px solid #007bff;

}

.tab-button:hover:not(.active) {

background-color: #3a3a3a; /\* خلفية أغمق عند التحويم \*/

color: #ffffff; /\* نص أبيض عند التحويم \*/

}

textarea {

width: calc(100% - 20px);

padding: 12px;

margin-bottom: 15px;

background-color: #2c2c2c; /\* خلفية داكنة لمربعات النص \*/

border: 1px solid #444; /\* حدود داكنة \*/

color: #e0e0e0; /\* نص فاتح \*/

border-radius: 8px;

font-size: 0.9em;

box-sizing: border-box;

outline: none;

transition: border-color 0.3s ease;

min-height: 120px;

resize: vertical;

}

textarea:focus {

border-color: #007bff; /\* حدود زرقاء عند التركيز \*/

}

button {

background-color: #007bff; /\* أزرق ساطع للأزرار \*/

color: white;

padding: 10px 20px;

border: none;

border-radius: 8px;

cursor: pointer;

font-size: 1em;

font-weight: bold;

transition: background-color 0.3s ease, transform 0.2s ease;

width: 100%;

box-sizing: border-box;

margin-top: 10px;

}

button:hover {

background-color: #0056b3; /\* أزرق أغمق عند التحويم \*/

transform: translateY(-2px);

}

button:active {

transform: translateY(0);

}

button:disabled {

background-color: #555555; /\* رمادي داكن للأزرار المعطلة \*/

cursor: not-allowed;

transform: none;

}

.result-section {

margin-top: 20px;

background: #2c2c2c; /\* خلفية داكنة لأقسام النتائج \*/

padding: 15px;

border-radius: 8px;

border: 1px solid #444; /\* حدود داكنة \*/

box-shadow: 0 2px 5px rgba(0,0,0,0.2); /\* ظل أغمق \*/

}

.result-section h3 {

margin-top: 0;

color: #7bbaff; /\* أزرق فاتح للعناوين الفرعية \*/

font-size: 1.2em;

border-bottom: 1px solid #333; /\* حدود داكنة \*/

padding-bottom: 10px;

margin-bottom: 15px;

}

.score-bar {

width: 100%;

background-color: #444; /\* خلفية داكنة لشريط التقييم \*/

border-radius: 5px;

overflow: hidden;

margin-bottom: 15px;

}

.score-fill {

height: 25px;

text-align: center;

color: white;

font-weight: bold;

line-height: 25px;

border-radius: 5px;

transition: width 0.5s ease-in-out, background-color 0.5s ease-in-out;

}

/\* ألوان شريط التقييم بناءً على النتيجة \*/

.score-fill[style\*="background-color: rgb(220, 53, 69)"] { background-color: #dc3545 !important; } /\* Red \*/

.score-fill[style\*="background-color: rgb(40, 167, 69)"] { background-color: #28a745 !important; } /\* Green \*/

.score-fill[style\*="background-color: rgb(255, 193, 7)"] { background-color: #ffc107 !important; } /\* Yellow \*/

.score-text {

margin-top: 5px;

font-size: 0.9em;

text-align: center;

color: #b0b0b0; /\* نص فاتح \*/

}

.feedback-item {

background-color: #3a3a3a; /\* خلفية داكنة لعناصر الملاحظات \*/

border-left: 4px solid #007bff; /\* حدود زرقاء \*/

padding: 10px;

margin-bottom: 10px;

border-radius: 5px;

color: #e0e0e0; /\* نص فاتح \*/

}

.feedback-item strong {

color: #7bbaff; /\* أزرق فاتح للعناوين داخل الملاحظات \*/

}

.feedback-item pre {

background-color: #4a4a4a !important; /\* خلفية أغمق لمقتطفات الكود \*/

color: #e0e0e0; /\* نص فاتح لمقتطفات الكود \*/

}

.loading-spinner {

display: none; /\* Hidden by default \*/

border: 4px solid #444; /\* Light grey \*/

border-top: 4px solid #007bff; /\* Blue \*/

border-radius: 50%;

width: 30px;

height: 30px;

animation: spin 1s linear infinite;

margin: 20px auto;

}

@keyframes spin {

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

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

}

.message-box {

padding: 10px;

margin-bottom: 15px;

border-radius: 5px;

font-size: 0.9em;

text-align: center;

display: none; /\* مخفي افتراضيا \*/

}

.message-box.success {

background-color: #28a74533; /\* أخضر فاتح شفاف \*/

color: #28a745; /\* أخضر \*/

border: 1px solid #28a745;

}

.message-box.error {

background-color: #dc354533; /\* أحمر فاتح شفاف \*/

color: #dc3545; /\* أحمر \*/

border: 1px solid #dc3545;

}

.message-box.info {

background-color: #007bff33; /\* أزرق فاتح شفاف \*/

color: #007bff; /\* أزرق \*/

border: 1px solid #007bff;

}

.confirmation-modal {

display: none;

position: fixed;

z-index: 1;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgba(0,0,0,0.6); /\* خلفية أغمق للمودال \*/

padding-top: 60px;

}

.modal-content {

background-color: #2c2c2c; /\* خلفية داكنة للمودال \*/

margin: 5% auto;

padding: 20px;

border: 1px solid #444; /\* حدود داكنة \*/

width: 80%;

max-width: 500px;

border-radius: 10px;

text-align: center;

color: #e0e0e0; /\* نص فاتح \*/

}

.modal-content button {

width: auto;

margin: 10px;

padding: 8px 15px;

display: inline-block;

}

.modal-content .cancel-button {

background-color: #6c757d;

}

.modal-content .cancel-button:hover {

background-color: #5a6268;

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 15px;

color: #e0e0e0; /\* نص فاتح للجداول \*/

}

table td {

padding: 8px;

border: 1px solid #444; /\* حدود داكنة للجداول \*/

}

table tr:nth-child(even) {

background-color: #3a3a3a; /\* خلفية أغمق للصفوف الزوجية \*/

}

table tr:nth-child(odd) {

background-color: #2c2c2c; /\* خلفية أفتح للصفوف الفردية \*/

}

table td[style\*="background-color: #f2f2f2"] {

background-color: #3a3a3a !important; /\* خلفية داكنة لرؤوس الخلايا \*/

font-weight: bold;

color: #7bbaff; /\* لون فاتح لرؤوس الخلايا \*/

}

table pre {

background: none !important; /\* إزالة خلفية الـ pre داخل الجدول \*/

color: #e0e0e0; /\* نص فاتح للكود داخل الجدول \*/

}

</style>

</head>

<body>

<div class="container">

<h2>✨ مساعد Apps Script الذكي</h2>

<div class="tabs">

<button class="tab-button active" onclick="openTab(event, 'docsTab')">📚 توثيق الدوال</button>

<button class="tab-button" onclick="openTab(event, 'codeReviewTab')">👨‍💻 مراجعة الكود</button>

<button class="tab-button" onclick="openTab(event, 'projectInsightsTab')">📈 تحليلات المشروع</button> <!-- New Tab -->

</div>

<div id="messageBox" class="message-box"></div>

<div id="loadingSpinner" class="loading-spinner"></div>

<!-- Docs Tab Content -->

<div id="docsTab" class="tab-content active">

<h3>البحث في دوال Apps Script</h3>

<input type="text" id="fnName" placeholder="أدخل اسم الدالة (مثال: SpreadsheetApp.getActiveSpreadsheet)">

<button onclick="searchDoc()">بحث</button>

<div id="docsResult" class="result-section">

<p style="text-align: center; color: #999;">ستظهر نتائج البحث هنا.</p>

</div>

</div>

<!-- Code Review Tab Content -->

<div id="codeReviewTab" class="tab-content">

<h3>مراجعة الكود وتحسينه</h3>

<textarea id="codeText" placeholder="الصق الكود هنا للمراجعة..."></textarea>

<button onclick="getCodeFromCell()">📥 جلب الكود من الخلية النشطة</button>

<button onclick="reviewCode()">✨ مراجعة الكود</button>

<button onclick="explainCode()">✨ شرح الكود</button>

<button onclick="generateUnitTests()">✨ توليد اختبارات الوحدة</button>

<button onclick="generateJSDoc()">✨ توليد JSDoc</button>

<button onclick="optimizePerformance()">✨ تحسين الأداء</button>

<button id="suggestFileButton" onclick="suggestFile()">📄 اقتراح ملف للكود</button>

<button id="proposeAndStageButton" onclick="confirmProposeAndStage()" style="display: none;">🚀 تجهيز الكود في ورشة العمل</button>

<button id="applyPatchButton" onclick="confirmApplyPatch()" style="display: none;">✅ تطبيق التعديلات</button>

<div id="codeReviewResults" class="result-section" style="display: none;">

<h3>نتائج المراجعة</h3>

<div class="score-bar">

<div id="scoreFill" class="score-fill"></div>

</div>

<div id="scoreText" class="score-text"></div>

<p id="summary"></p>

<div id="feedbackItems"></div>

<div id="patchDisplay" style="margin-top: 15px; display: none;">

<h4>التعديلات المقترحة (Diff):</h4>

<pre id="patchContent" style="background-color: #4a4a4a; padding: 10px; border-radius: 5px; overflow-x: auto; font-family: monospace; font-size: 0.85em; color: #e0e0e0;"></pre>

</div>

</div>

<div id="fileSuggestionResult" class="result-section" style="display: none;">

<h3>اقتراح الملف</h3>

<p id="suggestedFileName"></p>

</div>

<div id="geminiExplanationResult" class="result-section" style="display: none;">

<h3>استجابة Gemini</h3>

<pre id="geminiExplanationContent" style="background-color: #4a4a4a; padding: 10px; border-radius: 5px; overflow-x: auto; font-family: monospace; font-size: 0.85em; color: #e0e0e0;"></pre>

</div>

</div>

<!-- New Project Insights Tab Content -->

<div id="projectInsightsTab" class="tab-content">

<h3>البحث الشامل واقتراحات المشروع</h3>

<textarea id="projectQuery" placeholder="اكتب استعلامك للبحث في ملفات المشروع (مثال: ابحث عن جميع الدوال التي لا تحتوي على تعليقات JSDoc، أو اقترح تحسينات أمنية في المشروع)..."></textarea>

<button onclick="analyzeProject()">🔍 بحث شامل</button>

<div id="projectInsightsResults" class="result-section" style="display: none;">

<h3>نتائج البحث والاقتراحات</h3>

<p id="projectInsightsSummary"></p>

<div id="projectSuggestions"></div>

</div>

</div>

</div>

<!-- Confirmation Modal -->

<div id="confirmationModal" class="confirmation-modal">

<div class="modal-content">

<p id="modalMessage"></p>

<button id="confirmYes">نعم</button>

<button id="confirmNo" class="cancel-button">إلغاء</button>

</div>

</div>

<script>

const messageBox = document.getElementById('messageBox');

const loadingSpinner = document.getElementById('loadingSpinner');

// Docs Tab Elements

const fnNameInput = document.getElementById('fnName');

const docsResultDiv = document.getElementById('docsResult');

// Code Review Tab Elements

const codeTextarea = document.getElementById('codeText');

const codeReviewResultsDiv = document.getElementById('codeReviewResults');

const scoreFill = document.getElementById('scoreFill');

const scoreText = document.getElementById('scoreText');

const summaryPara = document.getElementById('summary');

const feedbackItemsDiv = document.getElementById('feedbackItems');

const suggestFileButton = document.getElementById('suggestFileButton');

const proposeAndStageButton = document.getElementById('proposeAndStageButton');

const applyPatchButton = document.getElementById('applyPatchButton');

const fileSuggestionResultDiv = document.getElementById('fileSuggestionResult');

const suggestedFileNamePara = document.getElementById('suggestedFileName');

const patchDisplayDiv = document.getElementById('patchDisplay');

const patchContentPre = document.getElementById('patchContent');

const geminiExplanationResultDiv = document.getElementById('geminiExplanationResult');

const geminiExplanationContentPre = document.getElementById('geminiExplanationContent');

let currentPatch = null; // لتخزين الـ patch المقترح

let currentSuggestedFile = null; // لتخزين اسم الملف المقترح

// Project Insights Tab Elements

const projectQueryTextarea = document.getElementById('projectQuery');

const projectInsightsResultsDiv = document.getElementById('projectInsightsResults');

const projectInsightsSummaryPara = document.getElementById('projectInsightsSummary');

const projectSuggestionsDiv = document.getElementById('projectSuggestions');

// Confirmation Modal Elements

const confirmationModal = document.getElementById('confirmationModal');

const modalMessage = document.getElementById('modalMessage');

const confirmYesButton = document.getElementById('confirmYes');

const confirmNoButton = document.getElementById('confirmNo');

let modalCallback = null; // لتخزين الدالة التي سيتم استدعاؤها عند التأكيد

/\*\*

\* @function displayMessage

\* @description

\* تعرض رسالة للمستخدم في مربع مخصص.

\* @param {string} message - الرسالة المراد عرضها.

\* @param {'info'|'error'|'success'} type - نوع الرسالة.

\*/

function displayMessage(message, type) {

messageBox.textContent = message;

messageBox.className = 'message-box ' + type;

messageBox.style.display = 'block';

setTimeout(() => {

messageBox.style.display = 'none';

}, 3000);

}

/\*\*

\* @function setLoadingState

\* @description

\* تتحكم في عرض مؤشر التحميل وحالة الأزرار.

\* @param {boolean} isLoading - true لعرض التحميل وتعطيل الأزرار، false لإخفائه وتفعيل الأزرار.

\*/

function setLoadingState(isLoading) {

const buttons = document.querySelectorAll('button');

buttons.forEach(button => {

// لا تعطل أزرار التبويبات

if (!button.classList.contains('tab-button')) {

button.disabled = isLoading;

}

});

loadingSpinner.style.display = isLoading ? 'block' : 'none';

}

/\*\*

\* @function openTab

\* @description

\* تفتح التبويب المحدد وتخفي الآخرين.

\* @param {Event} evt - حدث النقر.

\* @param {string} tabName - اسم التبويب المراد فتحه.

\*/

function openTab(evt, tabName) {

const tabContents = document.getElementsByClassName("tab-content");

for (let i = 0; i < tabContents.length; i++) {

tabContents[i].style.display = "none";

}

const tabButtons = document.getElementsByClassName("tab-button");

for (let i = 0; i < tabButtons.length; i++) {

tabButtons[i].className = tabButtons[i].className.replace(" active", "");

}

document.getElementById(tabName).style.display = "block";

evt.currentTarget.className += " active";

}

// افتح التبويب الافتراضي عند التحميل

document.addEventListener('DOMContentLoaded', () => {

document.querySelector('.tab-button.active').click();

});

// --- Docs Tab Functions ---

/\*\*

\* @function searchDoc

\* @description

\* تقوم بالبحث عن توثيق دالة Apps Script المحددة.

\*/

function searchDoc() {

const fn = fnNameInput.value.trim();

if (!fn) {

displayMessage("يرجى إدخال اسم دالة للبحث.", "error");

return;

}

docsResultDiv.innerHTML = '<p style="text-align: center; color: #999;">⏳ جاري البحث...</p>';

setLoadingState(true);

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && (res.table || res.text)) {

docsResultDiv.innerHTML = res.table || res.text;

displayMessage("✅ تم العثور على التوثيق.", "success");

} else {

docsResultDiv.innerHTML = "<p style='text-align: center; color: #999;'>لا توجد معلومات متاحة لهذه الدالة.</p>";

displayMessage("لا توجد معلومات متاحة لهذه الدالة.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

docsResultDiv.innerHTML = `<p style="color:red; text-align:center;">🚨 حدث خطأ: ${e.message}</p>`;

displayMessage(`🚨 خطأ في البحث: ${e.message}`, "error");

console.error("Error searching documentation:", e);

})

.getBuiltinFunctionDoc(fn);

}

// --- Code Review Tab Functions ---

/\*\*

\* @function getCodeFromCell

\* @description

\* يجلب الكود من الخلية النشطة في Google Sheet.

\*/

function getCodeFromCell() {

displayMessage("جاري جلب الكود من الخلية النشطة...", "info");

setLoadingState(true);

codeReviewResultsDiv.style.display = 'none';

fileSuggestionResultDiv.style.display = 'none';

proposeAndStageButton.style.display = 'none';

applyPatchButton.style.display = 'none';

geminiExplanationResultDiv.style.display = 'none';

currentPatch = null;

google.script.run

.withSuccessHandler(code => {

setLoadingState(false);

if (code) {

codeTextarea.value = code;

displayMessage("✅ تم جلب الكود بنجاح.", "success");

} else {

displayMessage("لا يوجد كود في الخلية النشطة أو أنها فارغة.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

displayMessage(`🚨 فشل جلب الكود: ${e.message}`, "error");

console.error("Error getting code from cell:", e);

})

.getCodeFromActiveCell();

}

/\*\*

\* @function reviewCode

\* @description

\* ترسل الكود للمراجعة بواسطة دالة جانب الخادم.

\*/

function reviewCode() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود للمراجعة.", "error");

return;

}

codeReviewResultsDiv.style.display = 'none';

fileSuggestionResultDiv.style.display = 'none';

proposeAndStageButton.style.display = 'none';

applyPatchButton.style.display = 'none';

geminiExplanationResultDiv.style.display = 'none';

currentPatch = null;

setLoadingState(true);

displayMessage("جاري مراجعة الكود...", "info");

google.script.run

.withSuccessHandler(handleReviewResult)

.withFailureHandler(handleReviewError)

.reviewCodeForSidebar(scriptText, 'general'); // نوع المراجعة يمكن أن يكون ديناميكيًا إذا أضفت خيارًا لذلك

}

/\*\*

\* @function handleReviewResult

\* @description

\* يعالج نتيجة مراجعة الكود ويعرضها في الواجهة.

\* @param {object} result - كائن النتيجة من جانب الخادم.

\*/

function handleReviewResult(result) {

setLoadingState(false);

codeReviewResultsDiv.style.display = 'block';

displayMessage("تمت المراجعة بنجاح!", "success");

if (result.error) {

handleReviewError(new Error(result.error));

return;

}

const score = result.overallScore || 0;

const summary = result.summary || "لا يوجد ملخص متاح.";

const feedbackItems = result.feedbackItems || [];

currentPatch = result.patch || null; // تخزين الـ patch

scoreFill.style.width = `${score}%`;

let scoreColor = '#dc3545'; // Red

if (score > 70) scoreColor = '#28a745'; // Green

else if (score > 40) scoreColor = '#ffc107'; // Yellow

scoreFill.style.backgroundColor = scoreColor;

scoreFill.textContent = `${score.toFixed(0)}%`;

scoreText.textContent = `التقييم العام: ${score.toFixed(0)}%`;

summaryPara.textContent = summary;

feedbackItemsDiv.innerHTML = '';

if (feedbackItems.length > 0) {

feedbackItems.forEach(item => {

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

div.className = 'feedback-item';

div.innerHTML = `<strong>${item.title || 'ملاحظة'}:</strong> ${item.description || ''}`;

if (item.codeSnippet) {

div.innerHTML += `<pre style="background-color: #4a4a4a; padding: 5px; margin-top: 5px; border-radius: 3px; overflow-x: auto; color: #e0e0e0;"><code>${item.codeSnippet}</code></pre>`;

}

feedbackItemsDiv.appendChild(div);

});

} else {

feedbackItemsDiv.innerHTML = '<p style="text-align: center; color: #b0b0b0;">لا توجد ملاحظات محددة.</p>';

}

// عرض الـ patch إذا كان موجودًا

if (currentPatch) {

patchDisplayDiv.style.display = 'block';

patchContentPre.textContent = currentPatch;

applyPatchButton.style.display = 'block';

} else {

patchDisplayDiv.style.display = 'none';

patchContentPre.textContent = '';

applyPatchButton.style.display = 'none';

}

}

/\*\*

\* @function handleReviewError

\* @description

\* يعالج أخطاء مراجعة الكود ويعرضها.

\* @param {Error} error - كائن الخطأ.

\*/

function handleReviewError(error) {

setLoadingState(false);

codeReviewResultsDiv.style.display = 'none';

displayMessage(`🚨 حدث خطأ أثناء المراجعة: ${error.message}`, "error");

console.error("Error during code review:", error);

}

/\*\*

\* @function explainCode

\* @description

\* ترسل الكود لنموذج Gemini لشرحه.

\*/

function explainCode() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود لشرحه.", "error");

return;

}

geminiExplanationResultDiv.style.display = 'block';

geminiExplanationContentPre.textContent = '⏳ جاري شرح الكود بواسطة Gemini...';

setLoadingState(true);

displayMessage("جاري شرح الكود...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'text' && res.content) {

geminiExplanationContentPre.textContent = res.content;

displayMessage("✅ تم شرح الكود بنجاح.", "success");

} else if (res && res.type === 'message') {

geminiExplanationContentPre.textContent = res.text;

displayMessage(res.text, res.level);

} else {

geminiExplanationContentPre.textContent = 'لم يتمكن Gemini من شرح الكود.';

displayMessage("فشل شرح الكود.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

geminiExplanationContentPre.textContent = `🚨 حدث خطأ: ${e.message}`;

displayMessage(`🚨 خطأ في شرح الكود: ${e.message}`, "error");

console.error("Error explaining code:", e);

})

.explainCodeWithGemini(scriptText);

}

/\*\*

\* @function generateUnitTests

\* @description

\* ترسل الكود لنموذج Gemini لتوليد اختبارات الوحدة.

\*/

function generateUnitTests() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود لتوليد اختبارات الوحدة.", "error");

return;

}

geminiExplanationResultDiv.style.display = 'block';

geminiExplanationContentPre.textContent = '⏳ جاري توليد اختبارات الوحدة بواسطة Gemini...';

setLoadingState(true);

displayMessage("جاري توليد اختبارات الوحدة...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'text' && res.content) {

geminiExplanationContentPre.textContent = res.content;

displayMessage("✅ تم توليد اختبارات الوحدة بنجاح.", "success");

} else if (res && res.type === 'message') {

geminiExplanationContentPre.textContent = res.text;

displayMessage(res.text, res.level);

} else {

geminiExplanationContentPre.textContent = 'لم يتمكن Gemini من توليد اختبارات الوحدة.';

displayMessage("فشل توليد اختبارات الوحدة.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

geminiExplanationContentPre.textContent = `🚨 حدث خطأ: ${e.message}`;

displayMessage(`🚨 خطأ في توليد اختبارات الوحدة: ${e.message}`, "error");

console.error("Error generating unit tests:", e);

})

.generateUnitTestsWithGemini(scriptText);

}

/\*\*

\* @function generateJSDoc

\* @description

\* ترسل الكود لنموذج Gemini لتوليد تعليقات JSDoc له.

\*/

function generateJSDoc() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود لتوليد تعليقات JSDoc.", "error");

return;

}

geminiExplanationResultDiv.style.display = 'block';

geminiExplanationContentPre.textContent = '⏳ جاري توليد تعليقات JSDoc بواسطة Gemini...';

setLoadingState(true);

displayMessage("جاري توليد تعليقات JSDoc...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'text' && res.content) {

geminiExplanationContentPre.textContent = res.content;

displayMessage("✅ تم توليد تعليقات JSDoc بنجاح.", "success");

} else if (res && res.type === 'message') {

geminiExplanationContentPre.textContent = res.text;

displayMessage(res.text, res.level);

} else {

geminiExplanationContentPre.textContent = 'لم يتمكن Gemini من توليد تعليقات JSDoc.';

displayMessage("فشل توليد تعليقات JSDoc.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

geminiExplanationContentPre.textContent = `🚨 حدث خطأ: ${e.message}`;

displayMessage(`🚨 خطأ في توليد تعليقات JSDoc: ${e.message}`, "error");

console.error("Error generating JSDoc:", e);

})

.generateJSDocWithGemini(scriptText);

}

/\*\*

\* @function optimizePerformance

\* @description

\* ترسل الكود لنموذج Gemini لاقتراح تحسينات الأداء.

\*/

function optimizePerformance() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود لاقتراح تحسينات الأداء.", "error");

return;

}

geminiExplanationResultDiv.style.display = 'block';

geminiExplanationContentPre.textContent = '⏳ جاري اقتراح تحسينات الأداء بواسطة Gemini...';

setLoadingState(true);

displayMessage("جاري اقتراح تحسينات الأداء...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'text' && res.content) {

geminiExplanationContentPre.textContent = res.content;

displayMessage("✅ تم اقتراح تحسينات الأداء بنجاح.", "success");

} else if (res && res.type === 'message') {

geminiExplanationContentPre.textContent = res.text;

displayMessage(res.text, res.level);

} else {

geminiExplanationContentPre.textContent = 'لم يتمكن Gemini من اقتراح تحسينات الأداء.';

displayMessage("فشل اقتراح تحسينات الأداء.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

geminiExplanationContentPre.textContent = `🚨 حدث خطأ: ${e.message}`;

displayMessage(`🚨 خطأ في اقتراح تحسينات الأداء: ${e.message}`, "error");

console.error("Error optimizing performance:", e);

})

.optimizePerformanceWithGemini(scriptText);

}

/\*\*

\* @function suggestFile

\* @description

\* ترسل الكود لاقتراح ملف مناسب له.

\*/

function suggestFile() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود لاقتراح ملف.", "error");

return;

}

fileSuggestionResultDiv.style.display = 'none';

proposeAndStageButton.style.display = 'none';

setLoadingState(true);

displayMessage("جاري اقتراح ملف...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.suggestedFile) {

currentSuggestedFile = res.suggestedFile;

fileSuggestionResultDiv.style.display = 'block';

suggestedFileNamePara.innerHTML = `الملف المقترح: <strong>${res.suggestedFile}</strong>`;

proposeAndStageButton.style.display = 'block';

displayMessage("✅ تم اقتراح الملف بنجاح.", "success");

} else {

fileSuggestionResultDiv.style.display = 'block';

suggestedFileNamePara.textContent = "لم يتم العثور على اقتراح لملف مناسب.";

displayMessage("لا يوجد اقتراح لملف.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

displayMessage(`🚨 فشل اقتراح الملف: ${e.message}`, "error");

console.error("Error suggesting file:", e);

})

.suggestFileFromSidebar(scriptText);

}

/\*\*

\* @function confirmProposeAndStage

\* @description

\* تعرض مربع تأكيد قبل تجهيز الكود في ورشة العمل.

\*/

function confirmProposeAndStage() {

if (!codeTextarea.value.trim() || !currentSuggestedFile) {

displayMessage("الرجاء جلب الكود واقتراح ملف أولاً.", "error");

return;

}

showConfirmationModal(`هل أنت متأكد أنك تريد تجهيز هذا الكود في ملف "${currentSuggestedFile}" بورشة العمل؟`, proposeAndStage);

}

/\*\*

\* @function proposeAndStage

\* @description

\* ترسل الكود لتجهيزه في ورشة العمل (إنشاء وحدة جديدة أو تعديل).

\*/

function proposeAndStage() {

const scriptText = codeTextarea.value.trim();

setLoadingState(true);

displayMessage("جاري تجهيز الكود في ورشة العمل...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'message') {

displayMessage(res.text, res.level);

} else {

displayMessage("✅ تم تجهيز الكود بنجاح في ورشة العمل.", "success");

}

})

.withFailureHandler(e => {

setLoadingState(false);

displayMessage(`🚨 فشل تجهيز الكود: ${e.message}`, "error");

console.error("Error proposing and staging code:", e);

})

.proposeAndStageCode(scriptText, currentSuggestedFile);

}

/\*\*

\* @function confirmApplyPatch

\* @description

\* تعرض مربع تأكيد قبل تطبيق التعديلات.

\*/

function confirmApplyPatch() {

if (!currentPatch) {

displayMessage("لا توجد تعديلات لتطبيقها.", "error");

return;

}

showConfirmationModal("هل أنت متأكد أنك تريد تطبيق هذه التعديلات على الكود في الخلية النشطة؟", applyPatch);

}

/\*\*

\* @function applyPatch

\* @description

\* تطبق التعديلات (patch) على الكود في الخلية النشطة.

\*/

function applyPatch() {

if (!currentPatch) {

displayMessage("لا توجد تعديلات لتطبيقها.", "error");

return;

}

setLoadingState(true);

displayMessage("جاري تطبيق التعديلات...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'message' && res.level === 'success') {

displayMessage(res.text, 'success');

// يمكن تحديث codeTextarea.value هنا إذا تم إرجاع الكود الجديد

// أو ببساطة إعلام المستخدم بالنجاح.

} else if (res && res.type === 'message' && res.level === 'error') {

displayMessage(res.text, 'error');

} else {

displayMessage("✅ تم تطبيق التعديلات بنجاح.", "success");

}

currentPatch = null; // مسح الـ patch بعد التطبيق

patchDisplayDiv.style.display = 'none';

applyPatchButton.style.display = 'none';

})

.withFailureHandler(e => {

setLoadingState(false);

displayMessage(`🚨 فشل تطبيق التعديلات: ${e.message}`, "error");

console.error("Error applying patch:", e);

})

.updateActiveCellWithCode(currentPatch); // إرسال الـ patch لتطبيقه

}

// --- Project Insights Functions ---

/\*\*

\* @function analyzeProject

\* @description

\* ترسل استعلام البحث الشامل إلى نموذج Gemini لتحليل المشروع.

\*/

function analyzeProject() {

const query = projectQueryTextarea.value.trim();

if (!query) {

displayMessage("يرجى إدخال استعلام للبحث في المشروع.", "error");

return;

}

projectInsightsResultsDiv.style.display = 'block';

projectInsightsSummaryPara.textContent = '⏳ جاري تحليل المشروع بواسطة Gemini...';

projectSuggestionsDiv.innerHTML = '';

setLoadingState(true);

displayMessage("جاري البحث الشامل...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'project\_insights' && res.summary) {

projectInsightsSummaryPara.textContent = res.summary;

projectSuggestionsDiv.innerHTML = '';

if (res.suggestions && res.suggestions.length > 0) {

res.suggestions.forEach((s, index) => {

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

suggestionDiv.className = 'feedback-item'; // Reuse styling

suggestionDiv.innerHTML = `<strong>${s.description}</strong>`;

if (s.file) {

suggestionDiv.innerHTML += `<p>الملف: <code>${s.file}</code></p>`;

}

if (s.codeSnippet) {

suggestionDiv.innerHTML += `<pre style="background-color: #4a4a4a; padding: 5px; margin-top: 5px; border-radius: 3px; overflow-x: auto; color: #e0e0e0;"><code>${s.codeSnippet}</code></pre>`;

}

if (s.patch) {

const applyBtn = document.createElement('button');

applyBtn.textContent = `✅ تطبيق الاقتراح على ${s.file || 'الخلية النشطة'}`;

applyBtn.onclick = () => confirmApplyProjectPatch(s.file, s.patch);

applyBtn.style.marginTop = '10px';

applyBtn.style.width = 'fit-content';

applyBtn.style.padding = '8px 15px';

applyBtn.style.display = 'block';

suggestionDiv.appendChild(applyBtn);

}

projectSuggestionsDiv.appendChild(suggestionDiv);

});

displayMessage("✅ تم العثور على اقتراحات للمشروع.", "success");

} else {

projectSuggestionsDiv.innerHTML = '<p style="text-align: center; color: #b0b0b0;">لا توجد اقتراحات محددة لهذا الاستعلام.</p>';

displayMessage("لا توجد اقتراحات.", "info");

}

} else if (res && res.type === 'message') {

projectInsightsSummaryPara.textContent = res.text;

displayMessage(res.text, res.level);

} else {

projectInsightsSummaryPara.textContent = 'لم يتمكن Gemini من تحليل المشروع.';

displayMessage("فشل تحليل المشروع.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

projectInsightsSummaryPara.textContent = `🚨 حدث خطأ: ${e.message}`;

displayMessage(`🚨 خطأ في البحث الشامل: ${e.message}`, "error");

console.error("Error analyzing project:", e);

})

.analyzeProjectForSuggestions(query);

}

/\*\*

\* @function confirmApplyProjectPatch

\* @description

\* تعرض مربع تأكيد قبل تطبيق تعديل من تحليل المشروع.

\* @param {string} fileName - اسم الملف الذي سيتم تطبيق التعديل عليه.

\* @param {string} patchContent - محتوى الـ patch.

\*/

function confirmApplyProjectPatch(fileName, patchContent) {

showConfirmationModal(`هل أنت متأكد أنك تريد تطبيق هذا التعديل على ملف "${fileName || 'الخلية النشطة'}"؟`, () => applyProjectPatch(fileName, patchContent));

}

/\*\*

\* @function applyProjectPatch

\* @description

\* تطبق الـ patch المقترح من تحليل المشروع على الملف المحدد أو الخلية النشطة.

\* @param {string} fileName - اسم الملف الذي سيتم تطبيق التعديل عليه.

\* @param {string} patchContent - محتوى الـ patch.

\*/

function applyProjectPatch(fileName, patchContent) {

setLoadingState(true);

displayMessage(`جاري تطبيق التعديل على ${fileName || 'الخلية النشطة'}...`, "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'message' && res.level === 'success') {

displayMessage(res.text, 'success');

} else if (res && res.type === 'message' && res.level === 'error') {

displayMessage(res.text, 'error');

} else {

displayMessage("✅ تم تطبيق التعديل بنجاح.", "success");

}

})

.withFailureHandler(e => {

setLoadingState(false);

displayMessage(`🚨 فشل تطبيق التعديل: ${e.message}`, "error");

console.error("Error applying project patch:", e);

})

.applyProjectPatch(fileName, patchContent); // دالة جديدة في .gs

}

// --- Confirmation Modal Logic ---

function showConfirmationModal(message, callback) {

modalMessage.textContent = message;

confirmationModal.style.display = 'block';

modalCallback = callback;

}

confirmYesButton.onclick = () => {

confirmationModal.style.display = 'none';

if (modalCallback) {

modalCallback();

modalCallback = null;

}

};

confirmNoButton.onclick = () => {

confirmationModal.style.display = 'none';

modalCallback = null;

displayMessage("تم إلغاء العملية.", "info");

setLoadingState(false); // إعادة تفعيل الأزرار إذا كانت معطلة بسبب التأكيد

};

</script>

</body>

</html>

<!DOCTYPE html>

<html lang="ar" dir="rtl">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>مساعد Apps Script الذكي</title>

<style>

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

direction: rtl;

padding: 0;

margin: 0;

background-color: #1a1a1a; /\* خلفية داكنة جداً \*/

color: #e0e0e0; /\* لون نص فاتح \*/

}

.container {

padding: 20px;

}

h2 {

text-align: center;

color: #7bbaff; /\* أزرق فاتح للعناوين \*/

margin-bottom: 25px;

font-size: 1.8em;

}

.tabs {

display: flex;

justify-content: center;

margin-bottom: 20px;

border-bottom: 1px solid #333; /\* حدود داكنة \*/

}

.tab-button {

background-color: #2c2c2c; /\* خلفية أزرار التبويب غير النشطة \*/

border: none;

padding: 10px 15px;

cursor: pointer;

font-size: 1em;

font-weight: bold;

color: #b0b0b0; /\* لون نص أزرار التبويب غير النشطة \*/

border-radius: 5px 5px 0 0;

transition: background-color 0.3s ease, color 0.3s ease;

}

.tab-button.active {

background-color: #007bff; /\* أزرق ساطع للتبويب النشط \*/

color: white;

border-bottom: 2px solid #007bff;

}

.tab-button:hover:not(.active) {

background-color: #3a3a3a; /\* خلفية أغمق عند التحويم \*/

color: #ffffff; /\* نص أبيض عند التحويم \*/

}

textarea {

width: calc(100% - 20px);

padding: 12px;

margin-bottom: 15px;

background-color: #2c2c2c; /\* خلفية داكنة لمربعات النص \*/

border: 1px solid #444; /\* حدود داكنة \*/

color: #e0e0e0; /\* نص فاتح \*/

border-radius: 8px;

font-size: 0.9em;

box-sizing: border-box;

outline: none;

transition: border-color 0.3s ease;

min-height: 120px;

resize: vertical;

}

textarea:focus {

border-color: #007bff; /\* حدود زرقاء عند التركيز \*/

}

button {

background-color: #007bff; /\* أزرق ساطع للأزرار \*/

color: white;

padding: 10px 20px;

border: none;

border-radius: 8px;

cursor: pointer;

font-size: 1em;

font-weight: bold;

transition: background-color 0.3s ease, transform 0.2s ease;

width: 100%;

box-sizing: border-box;

margin-top: 10px;

}

button:hover {

background-color: #0056b3; /\* أزرق أغمق عند التحويم \*/

transform: translateY(-2px);

}

button:active {

transform: translateY(0);

}

button:disabled {

background-color: #555555; /\* رمادي داكن للأزرار المعطلة \*/

cursor: not-allowed;

transform: none;

}

.result-section {

margin-top: 20px;

background: #2c2c2c; /\* خلفية داكنة لأقسام النتائج \*/

padding: 15px;

border-radius: 8px;

border: 1px solid #444; /\* حدود داكنة \*/

box-shadow: 0 2px 5px rgba(0,0,0,0.2); /\* ظل أغمق \*/

}

.result-section h3 {

margin-top: 0;

color: #7bbaff; /\* أزرق فاتح للعناوين الفرعية \*/

font-size: 1.2em;

border-bottom: 1px solid #333; /\* حدود داكنة \*/

padding-bottom: 10px;

margin-bottom: 15px;

}

.score-bar {

width: 100%;

background-color: #444; /\* خلفية داكنة لشريط التقييم \*/

border-radius: 5px;

overflow: hidden;

margin-bottom: 15px;

}

.score-fill {

height: 25px;

text-align: center;

color: white;

font-weight: bold;

line-height: 25px;

border-radius: 5px;

transition: width 0.5s ease-in-out, background-color 0.5s ease-in-out;

}

/\* ألوان شريط التقييم بناءً على النتيجة \*/

.score-fill[style\*="background-color: rgb(220, 53, 69)"] { background-color: #dc3545 !important; } /\* Red \*/

.score-fill[style\*="background-color: rgb(40, 167, 69)"] { background-color: #28a745 !important; } /\* Green \*/

.score-fill[style\*="background-color: rgb(255, 193, 7)"] { background-color: #ffc107 !important; } /\* Yellow \*/

.score-text {

margin-top: 5px;

font-size: 0.9em;

text-align: center;

color: #b0b0b0; /\* نص فاتح \*/

}

.feedback-item {

background-color: #3a3a3a; /\* خلفية داكنة لعناصر الملاحظات \*/

border-left: 4px solid #007bff; /\* حدود زرقاء \*/

padding: 10px;

margin-bottom: 10px;

border-radius: 5px;

color: #e0e0e0; /\* نص فاتح \*/

}

.feedback-item strong {

color: #7bbaff; /\* أزرق فاتح للعناوين داخل الملاحظات \*/

}

.feedback-item pre {

background-color: #4a4a4a !important; /\* خلفية أغمق لمقتطفات الكود \*/

color: #e0e0e0; /\* نص فاتح لمقتطفات الكود \*/

}

.loading-spinner {

display: none; /\* Hidden by default \*/

border: 4px solid #444; /\* Light grey \*/

border-top: 4px solid #007bff; /\* Blue \*/

border-radius: 50%;

width: 30px;

height: 30px;

animation: spin 1s linear infinite;

margin: 20px auto;

}

@keyframes spin {

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

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

}

.message-box {

padding: 10px;

margin-bottom: 15px;

border-radius: 5px;

font-size: 0.9em;

text-align: center;

display: none; /\* مخفي افتراضيا \*/

}

.message-box.success {

background-color: #28a74533; /\* أخضر فاتح شفاف \*/

color: #28a745; /\* أخضر \*/

border: 1px solid #28a745;

}

.message-box.error {

background-color: #dc354533; /\* أحمر فاتح شفاف \*/

color: #dc3545; /\* أحمر \*/

border: 1px solid #dc3545;

}

.message-box.info {

background-color: #007bff33; /\* أزرق فاتح شفاف \*/

color: #007bff; /\* أزرق \*/

border: 1px solid #007bff;

}

.confirmation-modal {

display: none;

position: fixed;

z-index: 1;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgba(0,0,0,0.6); /\* خلفية أغمق للمودال \*/

padding-top: 60px;

}

.modal-content {

background-color: #2c2c2c; /\* خلفية داكنة للمودال \*/

margin: 5% auto;

padding: 20px;

border: 1px solid #444; /\* حدود داكنة \*/

width: 80%;

max-width: 500px;

border-radius: 10px;

text-align: center;

color: #e0e0e0; /\* نص فاتح \*/

}

.modal-content button {

width: auto;

margin: 10px;

padding: 8px 15px;

display: inline-block;

}

.modal-content .cancel-button {

background-color: #6c757d;

}

.modal-content .cancel-button:hover {

background-color: #5a6268;

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 15px;

color: #e0e0e0; /\* نص فاتح للجداول \*/

}

table td {

padding: 8px;

border: 1px solid #444; /\* حدود داكنة للجداول \*/

}

table tr:nth-child(even) {

background-color: #3a3a3a; /\* خلفية أغمق للصفوف الزوجية \*/

}

table tr:nth-child(odd) {

background-color: #2c2c2c; /\* خلفية أفتح للصفوف الفردية \*/

}

table td[style\*="background-color: #f2f2f2"] {

background-color: #3a3a3a !important; /\* خلفية داكنة لرؤوس الخلايا \*/

font-weight: bold;

color: #7bbaff; /\* لون فاتح لرؤوس الخلايا \*/

}

table pre {

background: none !important; /\* إزالة خلفية الـ pre داخل الجدول \*/

color: #e0e0e0; /\* نص فاتح للكود داخل الجدول \*/

}

</style>

</head>

<body>

<div class="container">

<h2>✨ مساعد Apps Script الذكي</h2>

<div class="tabs">

<button class="tab-button active" onclick="openTab(event, 'docsTab')">📚 توثيق الدوال</button>

<button class="tab-button" onclick="openTab(event, 'codeReviewTab')">👨‍💻 مراجعة الكود</button>

<button class="tab-button" onclick="openTab(event, 'projectInsightsTab')">📈 تحليلات المشروع</button> <!-- New Tab -->

</div>

<div id="messageBox" class="message-box"></div>

<div id="loadingSpinner" class="loading-spinner"></div>

<!-- Docs Tab Content -->

<div id="docsTab" class="tab-content active">

<h3>البحث في دوال Apps Script</h3>

<input type="text" id="fnName" placeholder="أدخل اسم الدالة (مثال: SpreadsheetApp.getActiveSpreadsheet)">

<button onclick="searchDoc()">بحث</button>

<div id="docsResult" class="result-section">

<p style="text-align: center; color: #999;">ستظهر نتائج البحث هنا.</p>

</div>

</div>

<!-- Code Review Tab Content -->

<div id="codeReviewTab" class="tab-content">

<h3>مراجعة الكود وتحسينه</h3>

<textarea id="codeText" placeholder="الصق الكود هنا للمراجعة..."></textarea>

<button onclick="getCodeFromCell()">📥 جلب الكود من الخلية النشطة</button>

<button onclick="reviewCode()">✨ مراجعة الكود</button>

<button onclick="explainCode()">✨ شرح الكود</button>

<button onclick="generateUnitTests()">✨ توليد اختبارات الوحدة</button>

<button onclick="generateJSDoc()">✨ توليد JSDoc</button>

<button onclick="optimizePerformance()">✨ تحسين الأداء</button>

<button id="suggestFileButton" onclick="suggestFile()">📄 اقتراح ملف للكود</button>

<button id="proposeAndStageButton" onclick="confirmProposeAndStage()" style="display: none;">🚀 تجهيز الكود في ورشة العمل</button>

<button id="applyPatchButton" onclick="confirmApplyPatch()" style="display: none;">✅ تطبيق التعديلات</button>

<div id="codeReviewResults" class="result-section" style="display: none;">

<h3>نتائج المراجعة</h3>

<div class="score-bar">

<div id="scoreFill" class="score-fill"></div>

</div>

<div id="scoreText" class="score-text"></div>

<p id="summary"></p>

<div id="feedbackItems"></div>

<div id="patchDisplay" style="margin-top: 15px; display: none;">

<h4>التعديلات المقترحة (Diff):</h4>

<pre id="patchContent" style="background-color: #4a4a4a; padding: 10px; border-radius: 5px; overflow-x: auto; font-family: monospace; font-size: 0.85em; color: #e0e0e0;"></pre>

</div>

</div>

<div id="fileSuggestionResult" class="result-section" style="display: none;">

<h3>اقتراح الملف</h3>

<p id="suggestedFileName"></p>

</div>

<div id="geminiExplanationResult" class="result-section" style="display: none;">

<h3>استجابة Gemini</h3>

<pre id="geminiExplanationContent" style="background-color: #4a4a4a; padding: 10px; border-radius: 5px; overflow-x: auto; font-family: monospace; font-size: 0.85em; color: #e0e0e0;"></pre>

</div>

</div>

<!-- New Project Insights Tab Content -->

<div id="projectInsightsTab" class="tab-content">

<h3>البحث الشامل واقتراحات المشروع</h3>

<textarea id="projectQuery" placeholder="اكتب استعلامك للبحث في ملفات المشروع (مثال: ابحث عن جميع الدوال التي لا تحتوي على تعليقات JSDoc، أو اقترح تحسينات أمنية في المشروع)..."></textarea>

<button onclick="analyzeProject()">🔍 بحث شامل</button>

<div id="projectInsightsResults" class="result-section" style="display: none;">

<h3>نتائج البحث والاقتراحات</h3>

<p id="projectInsightsSummary"></p>

<div id="projectSuggestions"></div>

</div>

</div>

</div>

<!-- Confirmation Modal -->

<div id="confirmationModal" class="confirmation-modal">

<div class="modal-content">

<p id="modalMessage"></p>

<button id="confirmYes">نعم</button>

<button id="confirmNo" class="cancel-button">إلغاء</button>

</div>

</div>

<script>

const messageBox = document.getElementById('messageBox');

const loadingSpinner = document.getElementById('loadingSpinner');

// Docs Tab Elements

const fnNameInput = document.getElementById('fnName');

const docsResultDiv = document.getElementById('docsResult');

// Code Review Tab Elements

const codeTextarea = document.getElementById('codeText');

const codeReviewResultsDiv = document.getElementById('codeReviewResults');

const scoreFill = document.getElementById('scoreFill');

const scoreText = document.getElementById('scoreText');

const summaryPara = document.getElementById('summary');

const feedbackItemsDiv = document.getElementById('feedbackItems');

const suggestFileButton = document.getElementById('suggestFileButton');

const proposeAndStageButton = document.getElementById('proposeAndStageButton');

const applyPatchButton = document.getElementById('applyPatchButton');

const fileSuggestionResultDiv = document.getElementById('fileSuggestionResult');

const suggestedFileNamePara = document.getElementById('suggestedFileName');

const patchDisplayDiv = document.getElementById('patchDisplay');

const patchContentPre = document.getElementById('patchContent');

const geminiExplanationResultDiv = document.getElementById('geminiExplanationResult');

const geminiExplanationContentPre = document.getElementById('geminiExplanationContent');

let currentPatch = null; // لتخزين الـ patch المقترح

let currentSuggestedFile = null; // لتخزين اسم الملف المقترح

// Project Insights Tab Elements

const projectQueryTextarea = document.getElementById('projectQuery');

const projectInsightsResultsDiv = document.getElementById('projectInsightsResults');

const projectInsightsSummaryPara = document.getElementById('projectInsightsSummary');

const projectSuggestionsDiv = document.getElementById('projectSuggestions');

// Confirmation Modal Elements

const confirmationModal = document.getElementById('confirmationModal');

const modalMessage = document.getElementById('modalMessage');

const confirmYesButton = document.getElementById('confirmYes');

const confirmNoButton = document.getElementById('confirmNo');

let modalCallback = null; // لتخزين الدالة التي سيتم استدعاؤها عند التأكيد

/\*\*

\* @function displayMessage

\* @description

\* تعرض رسالة للمستخدم في مربع مخصص.

\* @param {string} message - الرسالة المراد عرضها.

\* @param {'info'|'error'|'success'} type - نوع الرسالة.

\*/

function displayMessage(message, type) {

messageBox.textContent = message;

messageBox.className = 'message-box ' + type;

messageBox.style.display = 'block';

setTimeout(() => {

messageBox.style.display = 'none';

}, 3000);

}

/\*\*

\* @function setLoadingState

\* @description

\* تتحكم في عرض مؤشر التحميل وحالة الأزرار.

\* @param {boolean} isLoading - true لعرض التحميل وتعطيل الأزرار، false لإخفائه وتفعيل الأزرار.

\*/

function setLoadingState(isLoading) {

const buttons = document.querySelectorAll('button');

buttons.forEach(button => {

// لا تعطل أزرار التبويبات

if (!button.classList.contains('tab-button')) {

button.disabled = isLoading;

}

});

loadingSpinner.style.display = isLoading ? 'block' : 'none';

}

/\*\*

\* @function openTab

\* @description

\* تفتح التبويب المحدد وتخفي الآخرين.

\* @param {Event} evt - حدث النقر.

\* @param {string} tabName - اسم التبويب المراد فتحه.

\*/

function openTab(evt, tabName) {

const tabContents = document.getElementsByClassName("tab-content");

for (let i = 0; i < tabContents.length; i++) {

tabContents[i].style.display = "none";

}

const tabButtons = document.getElementsByClassName("tab-button");

for (let i = 0; i < tabButtons.length; i++) {

tabButtons[i].className = tabButtons[i].className.replace(" active", "");

}

document.getElementById(tabName).style.display = "block";

evt.currentTarget.className += " active";

}

// افتح التبويب الافتراضي عند التحميل

document.addEventListener('DOMContentLoaded', () => {

document.querySelector('.tab-button.active').click();

});

// --- Docs Tab Functions ---

/\*\*

\* @function searchDoc

\* @description

\* تقوم بالبحث عن توثيق دالة Apps Script المحددة.

\*/

function searchDoc() {

const fn = fnNameInput.value.trim();

if (!fn) {

displayMessage("يرجى إدخال اسم دالة للبحث.", "error");

return;

}

docsResultDiv.innerHTML = '<p style="text-align: center; color: #999;">⏳ جاري البحث...</p>';

setLoadingState(true);

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && (res.table || res.text)) {

docsResultDiv.innerHTML = res.table || res.text;

displayMessage("✅ تم العثور على التوثيق.", "success");

} else {

docsResultDiv.innerHTML = "<p style='text-align: center; color: #999;'>لا توجد معلومات متاحة لهذه الدالة.</p>";

displayMessage("لا توجد معلومات متاحة لهذه الدالة.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

docsResultDiv.innerHTML = `<p style="color:red; text-align:center;">🚨 حدث خطأ: ${e.message}</p>`;

displayMessage(`🚨 خطأ في البحث: ${e.message}`, "error");

console.error("Error searching documentation:", e);

})

.getBuiltinFunctionDoc(fn);

}

// --- Code Review Tab Functions ---

/\*\*

\* @function getCodeFromCell

\* @description

\* يجلب الكود من الخلية النشطة في Google Sheet.

\*/

function getCodeFromCell() {

displayMessage("جاري جلب الكود من الخلية النشطة...", "info");

setLoadingState(true);

codeReviewResultsDiv.style.display = 'none';

fileSuggestionResultDiv.style.display = 'none';

proposeAndStageButton.style.display = 'none';

applyPatchButton.style.display = 'none';

geminiExplanationResultDiv.style.display = 'none';

currentPatch = null;

google.script.run

.withSuccessHandler(code => {

setLoadingState(false);

if (code) {

codeTextarea.value = code;

displayMessage("✅ تم جلب الكود بنجاح.", "success");

} else {

displayMessage("لا يوجد كود في الخلية النشطة أو أنها فارغة.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

displayMessage(`🚨 فشل جلب الكود: ${e.message}`, "error");

console.error("Error getting code from cell:", e);

})

.getCodeFromActiveCell();

}

/\*\*

\* @function reviewCode

\* @description

\* ترسل الكود للمراجعة بواسطة دالة جانب الخادم.

\*/

function reviewCode() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود للمراجعة.", "error");

return;

}

codeReviewResultsDiv.style.display = 'none';

fileSuggestionResultDiv.style.display = 'none';

proposeAndStageButton.style.display = 'none';

applyPatchButton.style.display = 'none';

geminiExplanationResultDiv.style.display = 'none';

currentPatch = null;

setLoadingState(true);

displayMessage("جاري مراجعة الكود...", "info");

google.script.run

.withSuccessHandler(handleReviewResult)

.withFailureHandler(handleReviewError)

.reviewCodeForSidebar(scriptText, 'general'); // نوع المراجعة يمكن أن يكون ديناميكيًا إذا أضفت خيارًا لذلك

}

/\*\*

\* @function handleReviewResult

\* @description

\* يعالج نتيجة مراجعة الكود ويعرضها في الواجهة.

\* @param {object} result - كائن النتيجة من جانب الخادم.

\*/

function handleReviewResult(result) {

setLoadingState(false);

codeReviewResultsDiv.style.display = 'block';

displayMessage("تمت المراجعة بنجاح!", "success");

if (result.error) {

handleReviewError(new Error(result.error));

return;

}

const score = result.overallScore || 0;

const summary = result.summary || "لا يوجد ملخص متاح.";

const feedbackItems = result.feedbackItems || [];

currentPatch = result.patch || null; // تخزين الـ patch

scoreFill.style.width = `${score}%`;

let scoreColor = '#dc3545'; // Red

if (score > 70) scoreColor = '#28a745'; // Green

else if (score > 40) scoreColor = '#ffc107'; // Yellow

scoreFill.style.backgroundColor = scoreColor;

scoreFill.textContent = `${score.toFixed(0)}%`;

scoreText.textContent = `التقييم العام: ${score.toFixed(0)}%`;

summaryPara.textContent = summary;

feedbackItemsDiv.innerHTML = '';

if (feedbackItems.length > 0) {

feedbackItems.forEach(item => {

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

div.className = 'feedback-item';

div.innerHTML = `<strong>${item.title || 'ملاحظة'}:</strong> ${item.description || ''}`;

if (item.codeSnippet) {

div.innerHTML += `<pre style="background-color: #4a4a4a; padding: 5px; margin-top: 5px; border-radius: 3px; overflow-x: auto; color: #e0e0e0;"><code>${item.codeSnippet}</code></pre>`;

}

feedbackItemsDiv.appendChild(div);

});

} else {

feedbackItemsDiv.innerHTML = '<p style="text-align: center; color: #b0b0b0;">لا توجد ملاحظات محددة.</p>';

}

// عرض الـ patch إذا كان موجودًا

if (currentPatch) {

patchDisplayDiv.style.display = 'block';

patchContentPre.textContent = currentPatch;

applyPatchButton.style.display = 'block';

} else {

patchDisplayDiv.style.display = 'none';

patchContentPre.textContent = '';

applyPatchButton.style.display = 'none';

}

}

/\*\*

\* @function handleReviewError

\* @description

\* يعالج أخطاء مراجعة الكود ويعرضها.

\* @param {Error} error - كائن الخطأ.

\*/

function handleReviewError(error) {

setLoadingState(false);

codeReviewResultsDiv.style.display = 'none';

displayMessage(`🚨 حدث خطأ أثناء المراجعة: ${error.message}`, "error");

console.error("Error during code review:", error);

}

/\*\*

\* @function explainCode

\* @description

\* ترسل الكود لنموذج Gemini لشرحه.

\*/

function explainCode() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود لشرحه.", "error");

return;

}

geminiExplanationResultDiv.style.display = 'block';

geminiExplanationContentPre.textContent = '⏳ جاري شرح الكود بواسطة Gemini...';

setLoadingState(true);

displayMessage("جاري شرح الكود...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'text' && res.content) {

geminiExplanationContentPre.textContent = res.content;

displayMessage("✅ تم شرح الكود بنجاح.", "success");

} else if (res && res.type === 'message') {

geminiExplanationContentPre.textContent = res.text;

displayMessage(res.text, res.level);

} else {

geminiExplanationContentPre.textContent = 'لم يتمكن Gemini من شرح الكود.';

displayMessage("فشل شرح الكود.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

geminiExplanationContentPre.textContent = `🚨 حدث خطأ: ${e.message}`;

displayMessage(`🚨 خطأ في شرح الكود: ${e.message}`, "error");

console.error("Error explaining code:", e);

})

.explainCodeWithGemini(scriptText);

}

/\*\*

\* @function generateUnitTests

\* @description

\* ترسل الكود لنموذج Gemini لتوليد اختبارات الوحدة.

\*/

function generateUnitTests() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود لتوليد اختبارات الوحدة.", "error");

return;

}

geminiExplanationResultDiv.style.display = 'block';

geminiExplanationContentPre.textContent = '⏳ جاري توليد اختبارات الوحدة بواسطة Gemini...';

setLoadingState(true);

displayMessage("جاري توليد اختبارات الوحدة...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'text' && res.content) {

geminiExplanationContentPre.textContent = res.content;

displayMessage("✅ تم توليد اختبارات الوحدة بنجاح.", "success");

} else if (res && res.type === 'message') {

geminiExplanationContentPre.textContent = res.text;

displayMessage(res.text, res.level);

} else {

geminiExplanationContentPre.textContent = 'لم يتمكن Gemini من توليد اختبارات الوحدة.';

displayMessage("فشل توليد اختبارات الوحدة.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

geminiExplanationContentPre.textContent = `🚨 حدث خطأ: ${e.message}`;

displayMessage(`🚨 خطأ في توليد اختبارات الوحدة: ${e.message}`, "error");

console.error("Error generating unit tests:", e);

})

.generateUnitTestsWithGemini(scriptText);

}

/\*\*

\* @function generateJSDoc

\* @description

\* ترسل الكود لنموذج Gemini لتوليد تعليقات JSDoc له.

\*/

function generateJSDoc() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود لتوليد تعليقات JSDoc.", "error");

return;

}

geminiExplanationResultDiv.style.display = 'block';

geminiExplanationContentPre.textContent = '⏳ جاري توليد تعليقات JSDoc بواسطة Gemini...';

setLoadingState(true);

displayMessage("جاري توليد تعليقات JSDoc...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'text' && res.content) {

geminiExplanationContentPre.textContent = res.content;

displayMessage("✅ تم توليد تعليقات JSDoc بنجاح.", "success");

} else if (res && res.type === 'message') {

geminiExplanationContentPre.textContent = res.text;

displayMessage(res.text, res.level);

} else {

geminiExplanationContentPre.textContent = 'لم يتمكن Gemini من توليد تعليقات JSDoc.';

displayMessage("فشل توليد تعليقات JSDoc.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

geminiExplanationContentPre.textContent = `🚨 حدث خطأ: ${e.message}`;

displayMessage(`🚨 خطأ في توليد تعليقات JSDoc: ${e.message}`, "error");

console.error("Error generating JSDoc:", e);

})

.generateJSDocWithGemini(scriptText);

}

/\*\*

\* @function optimizePerformance

\* @description

\* ترسل الكود لنموذج Gemini لاقتراح تحسينات الأداء.

\*/

function optimizePerformance() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود لاقتراح تحسينات الأداء.", "error");

return;

}

geminiExplanationResultDiv.style.display = 'block';

geminiExplanationContentPre.textContent = '⏳ جاري اقتراح تحسينات الأداء بواسطة Gemini...';

setLoadingState(true);

displayMessage("جاري اقتراح تحسينات الأداء...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'text' && res.content) {

geminiExplanationContentPre.textContent = res.content;

displayMessage("✅ تم اقتراح تحسينات الأداء بنجاح.", "success");

} else if (res && res.type === 'message') {

geminiExplanationContentPre.textContent = res.text;

displayMessage(res.text, res.level);

} else {

geminiExplanationContentPre.textContent = 'لم يتمكن Gemini من اقتراح تحسينات الأداء.';

displayMessage("فشل اقتراح تحسينات الأداء.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

geminiExplanationContentPre.textContent = `🚨 حدث خطأ: ${e.message}`;

displayMessage(`🚨 خطأ في اقتراح تحسينات الأداء: ${e.message}`, "error");

console.error("Error optimizing performance:", e);

})

.optimizePerformanceWithGemini(scriptText);

}

/\*\*

\* @function suggestFile

\* @description

\* ترسل الكود لاقتراح ملف مناسب له.

\*/

function suggestFile() {

const scriptText = codeTextarea.value.trim();

if (!scriptText) {

displayMessage("يرجى إدخال بعض الكود لاقتراح ملف.", "error");

return;

}

fileSuggestionResultDiv.style.display = 'none';

proposeAndStageButton.style.display = 'none';

setLoadingState(true);

displayMessage("جاري اقتراح ملف...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.suggestedFile) {

currentSuggestedFile = res.suggestedFile;

fileSuggestionResultDiv.style.display = 'block';

suggestedFileNamePara.innerHTML = `الملف المقترح: <strong>${res.suggestedFile}</strong>`;

proposeAndStageButton.style.display = 'block';

displayMessage("✅ تم اقتراح الملف بنجاح.", "success");

} else {

fileSuggestionResultDiv.style.display = 'block';

suggestedFileNamePara.textContent = "لم يتم العثور على اقتراح لملف مناسب.";

displayMessage("لا يوجد اقتراح لملف.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

displayMessage(`🚨 فشل اقتراح الملف: ${e.message}`, "error");

console.error("Error suggesting file:", e);

})

.suggestFileFromSidebar(scriptText);

}

/\*\*

\* @function confirmProposeAndStage

\* @description

\* تعرض مربع تأكيد قبل تجهيز الكود في ورشة العمل.

\*/

function confirmProposeAndStage() {

if (!codeTextarea.value.trim() || !currentSuggestedFile) {

displayMessage("الرجاء جلب الكود واقتراح ملف أولاً.", "error");

return;

}

showConfirmationModal(`هل أنت متأكد أنك تريد تجهيز هذا الكود في ملف "${currentSuggestedFile}" بورشة العمل؟`, proposeAndStage);

}

/\*\*

\* @function proposeAndStage

\* @description

\* ترسل الكود لتجهيزه في ورشة العمل (إنشاء وحدة جديدة أو تعديل).

\*/

function proposeAndStage() {

const scriptText = codeTextarea.value.trim();

setLoadingState(true);

displayMessage("جاري تجهيز الكود في ورشة العمل...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'message') {

displayMessage(res.text, res.level);

} else {

displayMessage("✅ تم تجهيز الكود بنجاح في ورشة العمل.", "success");

}

})

.withFailureHandler(e => {

setLoadingState(false);

displayMessage(`🚨 فشل تجهيز الكود: ${e.message}`, "error");

console.error("Error proposing and staging code:", e);

})

.proposeAndStageCode(scriptText, currentSuggestedFile);

}

/\*\*

\* @function confirmApplyPatch

\* @description

\* تعرض مربع تأكيد قبل تطبيق التعديلات.

\*/

function confirmApplyPatch() {

if (!currentPatch) {

displayMessage("لا توجد تعديلات لتطبيقها.", "error");

return;

}

showConfirmationModal("هل أنت متأكد أنك تريد تطبيق هذه التعديلات على الكود في الخلية النشطة؟", applyPatch);

}

/\*\*

\* @function applyPatch

\* @description

\* تطبق التعديلات (patch) على الكود في الخلية النشطة.

\*/

function applyPatch() {

if (!currentPatch) {

displayMessage("لا توجد تعديلات لتطبيقها.", "error");

return;

}

setLoadingState(true);

displayMessage("جاري تطبيق التعديلات...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'message' && res.level === 'success') {

displayMessage(res.text, 'success');

// يمكن تحديث codeTextarea.value هنا إذا تم إرجاع الكود الجديد

// أو ببساطة إعلام المستخدم بالنجاح.

} else if (res && res.type === 'message' && res.level === 'error') {

displayMessage(res.text, 'error');

} else {

displayMessage("✅ تم تطبيق التعديلات بنجاح.", "success");

}

currentPatch = null; // مسح الـ patch بعد التطبيق

patchDisplayDiv.style.display = 'none';

applyPatchButton.style.display = 'none';

})

.withFailureHandler(e => {

setLoadingState(false);

displayMessage(`🚨 فشل تطبيق التعديلات: ${e.message}`, "error");

console.error("Error applying patch:", e);

})

.updateActiveCellWithCode(currentPatch); // إرسال الـ patch لتطبيقه

}

// --- Project Insights Functions ---

/\*\*

\* @function analyzeProject

\* @description

\* ترسل استعلام البحث الشامل إلى نموذج Gemini لتحليل المشروع.

\*/

function analyzeProject() {

const query = projectQueryTextarea.value.trim();

if (!query) {

displayMessage("يرجى إدخال استعلام للبحث في المشروع.", "error");

return;

}

projectInsightsResultsDiv.style.display = 'block';

projectInsightsSummaryPara.textContent = '⏳ جاري تحليل المشروع بواسطة Gemini...';

projectSuggestionsDiv.innerHTML = '';

setLoadingState(true);

displayMessage("جاري البحث الشامل...", "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'project\_insights' && res.summary) {

projectInsightsSummaryPara.textContent = res.summary;

projectSuggestionsDiv.innerHTML = '';

if (res.suggestions && res.suggestions.length > 0) {

res.suggestions.forEach((s, index) => {

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

suggestionDiv.className = 'feedback-item'; // Reuse styling

suggestionDiv.innerHTML = `<strong>${s.description}</strong>`;

if (s.file) {

suggestionDiv.innerHTML += `<p>الملف: <code>${s.file}</code></p>`;

}

if (s.codeSnippet) {

suggestionDiv.innerHTML += `<pre style="background-color: #4a4a4a; padding: 5px; margin-top: 5px; border-radius: 3px; overflow-x: auto; color: #e0e0e0;"><code>${s.codeSnippet}</code></pre>`;

}

if (s.patch) {

const applyBtn = document.createElement('button');

applyBtn.textContent = `✅ تطبيق الاقتراح على ${s.file || 'الخلية النشطة'}`;

applyBtn.onclick = () => confirmApplyProjectPatch(s.file, s.patch);

applyBtn.style.marginTop = '10px';

applyBtn.style.width = 'fit-content';

applyBtn.style.padding = '8px 15px';

applyBtn.style.display = 'block';

suggestionDiv.appendChild(applyBtn);

}

projectSuggestionsDiv.appendChild(suggestionDiv);

});

displayMessage("✅ تم العثور على اقتراحات للمشروع.", "success");

} else {

projectSuggestionsDiv.innerHTML = '<p style="text-align: center; color: #b0b0b0;">لا توجد اقتراحات محددة لهذا الاستعلام.</p>';

displayMessage("لا توجد اقتراحات.", "info");

}

} else if (res && res.type === 'message') {

projectInsightsSummaryPara.textContent = res.text;

displayMessage(res.text, res.level);

} else {

projectInsightsSummaryPara.textContent = 'لم يتمكن Gemini من تحليل المشروع.';

displayMessage("فشل تحليل المشروع.", "info");

}

})

.withFailureHandler(e => {

setLoadingState(false);

projectInsightsSummaryPara.textContent = `🚨 حدث خطأ: ${e.message}`;

displayMessage(`🚨 خطأ في البحث الشامل: ${e.message}`, "error");

console.error("Error analyzing project:", e);

})

.analyzeProjectForSuggestions(query);

}

/\*\*

\* @function confirmApplyProjectPatch

\* @description

\* تعرض مربع تأكيد قبل تطبيق تعديل من تحليل المشروع.

\* @param {string} fileName - اسم الملف الذي سيتم تطبيق التعديل عليه.

\* @param {string} patchContent - محتوى الـ patch.

\*/

function confirmApplyProjectPatch(fileName, patchContent) {

showConfirmationModal(`هل أنت متأكد أنك تريد تطبيق هذا التعديل على ملف "${fileName || 'الخلية النشطة'}"؟`, () => applyProjectPatch(fileName, patchContent));

}

/\*\*

\* @function applyProjectPatch

\* @description

\* تطبق الـ patch المقترح من تحليل المشروع على الملف المحدد أو الخلية النشطة.

\* @param {string} fileName - اسم الملف الذي سيتم تطبيق التعديل عليه.

\* @param {string} patchContent - محتوى الـ patch.

\*/

function applyProjectPatch(fileName, patchContent) {

setLoadingState(true);

displayMessage(`جاري تطبيق التعديل على ${fileName || 'الخلية النشطة'}...`, "info");

google.script.run

.withSuccessHandler(res => {

setLoadingState(false);

if (res && res.type === 'message' && res.level === 'success') {

displayMessage(res.text, 'success');

} else if (res && res.type === 'message' && res.level === 'error') {

displayMessage(res.text, 'error');

} else {

displayMessage("✅ تم تطبيق التعديل بنجاح.", "success");

}

})

.withFailureHandler(e => {

setLoadingState(false);

displayMessage(`🚨 فشل تطبيق التعديل: ${e.message}`, "error");

console.error("Error applying project patch:", e);

})

.applyProjectPatch(fileName, patchContent); // دالة جديدة في .gs

}

// --- Confirmation Modal Logic ---

function showConfirmationModal(message, callback) {

modalMessage.textContent = message;

confirmationModal.style.display = 'block';

modalCallback = callback;

}

confirmYesButton.onclick = () => {

confirmationModal.style.display = 'none';

if (modalCallback) {

modalCallback();

modalCallback = null;

}

};

confirmNoButton.onclick = () => {

confirmationModal.style.display = 'none';

modalCallback = null;

displayMessage("تم إلغاء العملية.", "info");

setLoadingState(false); // إعادة تفعيل الأزرار إذا كانت معطلة بسبب التأكيد

};

</script>

</body>

</html>