<!DOCTYPE html>

<html>

<head>

<base target="\_top">

<style>

body {

font-family: 'Segoe UI', sans-serif;

background-color: #202124;

color: #e8eaed;

padding: 16px;

direction: rtl;

}

label, select, textarea {

display: block;

margin-bottom: 12px;

width: 100%;

}

select, textarea {

padding: 8px;

background-color: #2d2e30;

border: 1px solid #3c4043;

color: #e8eaed;

border-radius: 6px;

font-size: 14px;

}

textarea {

min-height: 120px;

font-family: monospace;

}

button {

padding: 10px 20px;

margin: 6px 4px 0 0;

background-color: #8ab4f8;

color: #202124;

border: none;

border-radius: 6px;

font-weight: bold;

cursor: pointer;

transition: background-color 0.3s ease;

}

button:hover {

background-color: #7aa3e5;

}

button:disabled {

background-color: #5a6a7a;

cursor: not-allowed;

}

#status {

margin-top: 10px;

font-size: 13px;

color: #aaa;

}

#geminiResponse {

margin-top: 20px;

padding: 15px;

background-color: #2d2e30;

border: 1px solid #3c4043;

border-radius: 6px;

min-height: 100px;

color: #f3f4f6;

white-space: pre-wrap; /\* للحفاظ على تنسيق النص من النموذج \*/

word-wrap: break-word; /\* لكسر الكلمات الطويلة \*/

}

.loading-spinner {

border: 4px solid #f3f3f3;

border-top: 4px solid #8ab4f8;

border-radius: 50%;

width: 20px;

height: 20px;

animation: spin 1s linear infinite;

margin: 10px auto;

display: none;

}

@keyframes spin {

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

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

}

</style>

</head>

<body>

<h2>📄 توليد Prompt بناء على الورقة</h2>

<div class="input-group">

<label for="taskType">نوع المهمة:</label>

<select id="taskType">

<option value="تحليل مالي">تحليل مالي</option>

<option value="تلخيص بيانات">تلخيص بيانات</option>

<option value="توليد كود">توليد كود</option>

<option value="مساعدة عامة">مساعدة عامة</option>

<option value="مراجعة كود">مراجعة كود</option>

<option value="استخراج معلومات">استخراج معلومات</option>

</select>

</div>

<div class="input-group">

<label for="userPrompt">سؤال أو طلب المستخدم:</label>

<textarea id="userPrompt" placeholder="اكتب سؤالك أو طلبك هنا..."></textarea>

</div>

<div class="input-group">

<label for="style">أسلوب الكتابة (اختياري):</label>

<select id="style">

<option value="">تلقائي</option>

<option value="تقني ومفصل">تقني ومفصل</option>

<option value="مباشر وواضح">مباشر وواضح</option>

<option value="إبداعي">إبداعي</option>

<option value="رسمي">رسمي</option>

</select>

</div>

<div class="input-group">

<label for="sheetFormat">تنسيق وصف الورقة:</label>

<select id="sheetFormat">

<option value="markdown">Markdown</option>

<option value="json">JSON</option>

<option value="none">لا شيء</option>

</select>

</div>

<button onclick="generatePrompt()">توليد Prompt</button>

<button onclick="copyPrompt()">نسخ Prompt</button>

<button id="sendToGeminiButton" onclick="sendToGemini()">✨ إرسال إلى Gemini</button>

<button id="optimizePromptButton" onclick="optimizePrompt()">✨ تحسين Prompt</button>

<button id="generateIdeasButton" onclick="generateIdeas()">✨ اقتراح أفكار</button>

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

<label for="finalPrompt">الـ Prompt النهائي:</label>

<textarea id="finalPrompt" readonly placeholder="سيظهر الـ Prompt هنا..."></textarea>

<div id="status"></div>

<h3>استجابة Gemini:</h3>

<div id="geminiResponse">

<p style="color: #aaa; text-align: center;">ستظهر استجابة نموذج Gemini هنا.</p>

</div>

<script>

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

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

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

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

const geminiResponseDiv = document.getElementById('geminiResponse');

const finalPromptTextarea = document.getElementById('finalPrompt');

function displayMessage(message, type = 'info') {

const statusElement = document.getElementById("status");

statusElement.textContent = message;

statusElement.style.color = type === 'error' ? 'red' : (type === 'success' ? 'green' : '#aaa');

}

function setButtonsLoadingState(isLoading) {

sendToGeminiButton.disabled = isLoading;

optimizePromptButton.disabled = isLoading;

generateIdeasButton.disabled = isLoading;

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

if (isLoading) {

sendToGeminiButton.textContent = "جاري الإرسال...";

optimizePromptButton.textContent = "جاري التحسين...";

generateIdeasButton.textContent = "جاري الاقتراح...";

} else {

sendToGeminiButton.textContent = "✨ إرسال إلى Gemini";

optimizePromptButton.textContent = "✨ تحسين Prompt";

generateIdeasButton.textContent = "✨ اقتراح أفكار";

}

}

function generatePrompt() {

const taskType = document.getElementById("taskType").value;

const userPrompt = document.getElementById("userPrompt").value.trim();

const style = document.getElementById("style").value;

const sheetFormat = document.getElementById("sheetFormat").value;

if (!userPrompt) {

displayMessage("يرجى كتابة السؤال المطلوب أولًا.", "error");

return;

}

displayMessage("جاري توليد Prompt...", "info");

setButtonsLoadingState(true); // Disable buttons during prompt generation

google.script.run

.withSuccessHandler(prompt => {

finalPromptTextarea.value = prompt;

displayMessage("✅ تم توليد prompt.", "success");

setButtonsLoadingState(false); // Enable buttons after prompt generation

})

.withFailureHandler(e => {

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

setButtonsLoadingState(false); // Enable buttons on error

})

.PROMPT\_ENGINE\_buildPrompt({ taskType, userPrompt, style, sheetFormat });

}

function copyPrompt() {

const prompt = finalPromptTextarea.value;

if (!prompt) {

displayMessage("لم يتم توليد أي Prompt.", "error");

return;

}

document.execCommand('copy');

displayMessage("📋 تم النسخ إلى الحافظة.", "success");

}

/\*\*

\* @function sendToGemini

\* @description

\* ترسل الـ Prompt النهائي إلى نموذج Gemini عبر دالة askGemini في جانب الخادم

\* وتعرض الاستجابة في الواجهة.

\*/

function sendToGemini() {

const prompt = finalPromptTextarea.value;

if (!prompt) {

displayMessage("لم يتم توليد أي Prompt.", "error");

return;

}

setButtonsLoadingState(true);

geminiResponseDiv.innerHTML = '<p style="color: #aaa; text-align: center;">جاري الحصول على استجابة Gemini...</p>';

displayMessage("جاري إرسال Prompt إلى Gemini...", "info");

google.script.run

.withSuccessHandler(res => {

setButtonsLoadingState(false);

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

geminiResponseDiv.innerHTML = res.content;

displayMessage("✅ تم استلام الرد من Gemini بنجاح.", "success");

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

geminiResponseDiv.innerHTML = `<p style="color: ${res.level === 'error' ? 'red' : 'green'}; text-align: center;">${res.text}</p>`;

displayMessage(res.text, res.level);

} else {

geminiResponseDiv.innerHTML = `<p style="color: #aaa; text-align: center;">لا توجد استجابة محددة من النموذج.</p>`;

displayMessage("لا توجد استجابة محددة من النموذج.", "info");

}

})

.withFailureHandler(e => {

setButtonsLoadingState(false);

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

displayMessage(`❌ خطأ في إرسال Prompt إلى Gemini: ${e.message}`, "error");

console.error("Error sending prompt to Gemini:", e);

})

.askGemini(prompt, "gemini-1.5-flash"); // استخدام نموذج flash افتراضي

}

/\*\*

\* @function optimizePrompt

\* @description

\* ترسل الـ Prompt النهائي إلى نموذج Gemini لطلب تحسينه.

\*/

function optimizePrompt() {

const currentPrompt = finalPromptTextarea.value;

if (!currentPrompt) {

displayMessage("يرجى توليد Prompt أولاً لتحسينه.", "error");

return;

}

const optimizationRequest = `قم بتحسين الـ Prompt التالي ليكون أكثر وضوحًا وفعالية في الحصول على استجابة دقيقة من نموذج لغوي كبير، مع الأخذ في الاعتبار أنه سيُستخدم لتحليل بيانات جداول البيانات: \n\n${currentPrompt}`;

setButtonsLoadingState(true);

geminiResponseDiv.innerHTML = '<p style="color: #aaa; text-align: center;">جاري تحسين الـ Prompt بواسطة Gemini...</p>';

displayMessage("جاري تحسين الـ Prompt...", "info");

google.script.run

.withSuccessHandler(res => {

setButtonsLoadingState(false);

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

geminiResponseDiv.innerHTML = `<strong>الـ Prompt المحسّن:</strong><br>${res.content}`;

finalPromptTextarea.value = res.content; // تحديث الـ Prompt النهائي بالنسخة المحسّنة

displayMessage("✅ تم تحسين الـ Prompt بنجاح.", "success");

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

geminiResponseDiv.innerHTML = `<p style="color: ${res.level === 'error' ? 'red' : 'green'}; text-align: center;">${res.text}</p>`;

displayMessage(res.text, res.level);

} else {

geminiResponseDiv.innerHTML = `<p style="color: #aaa; text-align: center;">لم يتمكن Gemini من تحسين الـ Prompt.</p>`;

displayMessage("فشل تحسين الـ Prompt.", "info");

}

})

.withFailureHandler(e => {

setButtonsLoadingState(false);

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

displayMessage(`❌ خطأ في تحسين Prompt: ${e.message}`, "error");

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

})

.askGemini(optimizationRequest, "gemini-1.5-pro"); // قد تحتاج إلى نموذج أقوى للتحسين

}

/\*\*

\* @function generateIdeas

\* @description

\* تطلب من نموذج Gemini اقتراح أفكار لـ Prompts/مهام بناءً على سياق ورقة العمل.

\*/

function generateIdeas() {

const ideaGenerationPrompt = `بناءً على سياق ورقة العمل الحالية (الموضح في وصف الورقة أدناه)، اقترح 5 أفكار مبتكرة لـ Prompts أو مهام تحليلية يمكنني القيام بها باستخدام نموذج لغوي كبير. ركز على المهام التي يمكن أن توفر رؤى قيمة أو أتمتة.`;

setButtonsLoadingState(true);

geminiResponseDiv.innerHTML = '<p style="color: #aaa; text-align: center;">جاري اقتراح أفكار بواسطة Gemini...</p>';

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

// هنا يجب أن نضمن أن وصف الورقة يتم تضمينه في askGemini

// بما أن askGemini في Code.gs يمكنها جلب سياق الورقة تلقائيًا،

// سنعتمد على ذلك.

google.script.run

.withSuccessHandler(res => {

setButtonsLoadingState(false);

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

geminiResponseDiv.innerHTML = `<strong>أفكار مقترحة:</strong><br>${res.content}`;

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

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

geminiResponseDiv.innerHTML = `<p style="color: ${res.level === 'error' ? 'red' : 'green'}; text-align: center;">${res.text}</p>`;

displayMessage(res.text, res.level);

} else {

geminiResponseDiv.innerHTML = `<p style="color: #aaa; text-align: center;">لم يتمكن Gemini من اقتراح أفكار.</p>`;

displayMessage("فشل اقتراح الأفكار.", "info");

}

})

.withFailureHandler(e => {

setButtonsLoadingState(false);

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

displayMessage(`❌ خطأ في اقتراح الأفكار: ${e.message}`, "error");

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

})

.askGemini(ideaGenerationPrompt, "gemini-1.5-pro"); // قد تحتاج إلى نموذج أقوى لتوليد الأفكار

}

</script>

</body>

</html>