وثيقة رسمية: خارطة تطوير مشروع G-Assistant

**الاسم:** خارطة الطريق النهائية لتطوير مشروع G-Assistant  
 **الإصدار:** v15.6  
 **التاريخ:** الخميس 11 يوليو 2025  
 **المنصة:** Google Apps Script + Gemini SDK

✅ الملخص التنفيذي

| **التصنيف** | **العدد** |
| --- | --- |
| عدد الملفات الكلي | 24 |
| ملفات محسّنة | 11 |
| ملفات جديدة بالكامل | 7 |
| ملفات مدموجة أو محذوفة | 6 |

**المعمارية:** وحدات مغلقة (IIFE)  
 **التقنيات:** التفكير الداخلي • الصور (inlineData) • إخراج مهيكل • function\_call  
 **ميزة خاصة:** اختيار تلقائي للنموذج + اقتراح التبديل عبر suggestModelSwitch()

🧰 المعايير العامة لجميع الملفات

**إضافة مثال عملي** لـ suggestModelSwitch في ملحق الواجهة (Sidebar.html).

**توثيق الإعدادات** (AUTO\_PROCEED) ووضعها في ENV.

المحافظة وتطوير التنسيق لسهولة الصيانة ومعرفة بقية المبرمجين للتطوير مستقبلا

| **البند** | **المعيار** |
| --- | --- |
| التغليف | كل وحدة بصيغة var MODULE = (function(...) {...})(...) |
| التوثيق | JSDoc لكل دالة + وصف موحد في رأس كل ملف |
| معالجة الأخطاء | عبر HELPERS.logOperation و AI\_TOOLS\_DIALOGUE.showError |
| التبعية | كل ملف يستورد فقط ما يحتاجه (ENV, HELPERS, ...) |
| الأمان | منع حفظ المفاتيح الحساسة (مثل token، apikey) في السياق |
| دعم Gemini | صور Base64 / URL + thinkingConfig + structuredOutput + caching |
| السياق | CacheService لحفظ systemPrompt والمحتوى الثابت |
| النموذج الذكي | دعم اختيار النموذج تلقائيًا بحسب تحليل الطلب |
| توصية التبديل | إيقاف مؤقت واقتراح نموذج أدق باستخدام suggestModelSwitch() |

🧠 آلية اختيار النموذج (Model Selection)

✋ يدوي (من واجهة المستخدم)

* المستخدم يختار النموذج يدويًا من القائمة (مثل: Flash – Pro – Vision)
* القيمة تُمرر إلى state.model مباشرة

🤖 تلقائي (من داخل النظام)

* باستخدام getPreferredGeminiModel(prompt) داخل ai\_config.gs
* يرجع كائن مثل:

{ model: 'gemini-1.5-pro', complexity: 'high' }

💡 اقتراح بديل بدل التنفيذ

* إذا كان complexity === 'high' و state.autoProceed !== true
* ai\_core.gs يرجع:
* AI\_TOOLS\_DIALOGUE.suggestModelSwitch('gemini-1.5-pro', message) يظهر في Sidebar.html كخيار تفاعلي

🗂️ هيكل المشروع النهائي

📁 /src (الملفات البرمجية):

1. env.gs – إعدادات النظام
2. helpers.gs – أدوات مساعدة عامة
3. ai\_config.gs – اختيار النموذج الذكي + إعدادات مفضلة
4. ai\_context.gs – إنشاء system prompt والسياق
5. ai\_ask.gs – استدعاء Gemini SDK + إعدادات التفكير
6. ai\_memory.gs – إدارة السياق وتخزين طويل الأجل
7. ai\_dispatcher.gs – pre/post processing للردود
8. ai\_core.gs – pipeline موحّد: ask → dispatch → enrich → respond
9. tools\_catalog.gs – تعريف جميع الأدوات المتاحة
10. tools\_context.gs – دوال السياق والتخزين المؤقت
11. tools\_dialogue.gs – عرض الرسائل والتوصيات
12. tools\_sheets.gs – أدوات أوراق Google
13. tools\_accounting.gs – أدوات المحاسبة والتحليل المالي
14. tools\_general.gs – النسخ الاحتياطية، PDF، التصدير
15. tools\_internal.gs – دوال داخلية للمساعد
16. builtinDocs.gs – توثيق الدوال
17. tools\_doc\_gen.gs – بناء التوثيق تلقائيًا
18. workshop.gs – مكتبة اختبار الأكواد والتفاعل
19. tests.gs – اختبارات للوحدات والتحليل
20. ui.gs – onOpen والقوائم الجانبية

📁 /html (ملفات الواجهة):

Sidebar.html – المحادثة والتفكير والنموذج

BuiltinDocsSidebar.html – عرض التوثيق الوظيفي

PromptGeneratorSidebar.html – توليد الأوامر الذكية

TestResultsSidebar.html – عرض نتائج الاختبارات

MainUI.html – واجهة موحدة لتجميع النماذج والخيارات

📋 جدول التعديلات المفصلة لكل ملف  
جدول الملفات (مرتّب حسب ترتيب Google Apps Script)

| **📌 ترتيب Google Apps Script** | **🛠️ ترتيب الصيانة** | **📂 الملف** | **📝 التغييرات المضافة** | **🆕 الحالة** |
| --- | --- | --- | --- | --- |
| 1 | 1 | env.gs | إعداد SETTINGS, TEMPLATES, نماذج الذكاء، الألوان، الكاش | 🆕 جديد |
|  |  |  |  |  |
| 2 | 2 | helpers.gs | دوال عامة + fetchImageAsBase64, تحسين logOperation | ✅ محسّن |
| 3 | 3 | ai\_config.gs | اختيار النموذج الذكي + دالة getPreferredGeminiModel() | 🆕 جديد |
| 4 | 4 | ai\_context.gs | systemPrompt, دور المستخدم, history, إدخال للسياق | ✅ محسّن |
| 5 | 5 | ai\_ask.gs | الاتصال بـ SDK + structuredOutput + التفكير + دعم الصور | ✅ محسّن |
| 6 | 6 | ai\_memory.gs | سياق الجلسة طويل المدى + cache + expiresAt | ✅ محسّن |
| 7 | 7 | ai\_dispatcher.gs | تحليل الطلبات + postProcessResult من ردود Gemini | 🆕 جديد |
| 8 | 8 | ai\_core.gs | Pipeline كامل + دعم suggestModelSwitch حسب التعقيد | ✅ محسّن |
| 9 | 9 | tools\_catalog.gs | تصدير functionDeclarations من الأدوات لـ Gemini | ✅ محسّن |
| 10 | 10 | tools\_context.gs | فصل كامل لدوال setContext, summarizeSession, rememberFor | 🆕 جديد |
| 11 | 11 | tools\_dialogue.gs | رسائل المستخدم، suggestNextActions, suggestModelSwitch, showSummary | ✅ محسّن |
| 12 | 12 | tools\_sheets.gs | CRUD للجداول, filter, highlightEmptyCells, extractURLs | ✅ محسّن |
| 13 | 13 | tools\_accounting.gs | generateIncomeStatement, calculateGrossProfit, تحليل التصنيفات | ✅ محسّن |
| 14 | 14 | tools\_general.gs | نسخ احتياطية, export PDF, قيود المستخدمين | ✅ محسّن |
| 15 | 15 | tools\_internal.gs | دوال خاصة للمساعد: \_internal\_userConfirmed, دمج أوراق, إنشاء جداول | ✅ محسّن |
| 16 | 16 | builtinDocs.gs | تحليل أدوات المشروع وتوليد التوثيق المضمّن | 🆕 جديد |
| 17 | 17 | tools\_doc\_gen.gs | إنشاء وثائق تلقائيًا من الأدوات المسجلة | 🆕 جديد |
| 18 | 18 | workshop.gs | وحدة اختبار الأوامر والنماذج قيد التطوير | 🆕 جديد |
| 19 | 19 | tests.gs | اختبار التفكير، الصور، الكاش، توصيات النماذج | 🆕 جديد |
| 20 | 20 | ui.gs | onOpen موحد, تشغيل Sidebars, onInstall, قوائم أدوات متعددة | ✅ محسّن |

ملفات الواجهة (بترتيب التثبيت داخل المشروع):

| **📌 ترتيب Google Apps Script** | **🛠️ ترتيب الصيانة** | **📂 الملف** | **📝 التغييرات المضافة** | **🆕 الحالة** |
| --- | --- | --- | --- | --- |
| 21 | 21 | Sidebar.html | خيارات: إدخال صورة، اختيار النموذج، suggestModelSwitch, تفكير، بث مباشر | ✅ محسّن |
| 22 | 22 | BuiltinDocsSidebar.html | عرض توثيق الأدوات المستخرجة تلقائيًا من TOOLS\_CATALOG | ✅ محسّن |
| 23 | 23 | PromptGeneratorSidebar.html | توليد أوامر ديناميكية بناءً على التصنيف والوصف | 🆕 جديد |
| 24 | 24 | TestResultsSidebar.html | عرض نتائج اختبار أدوات التفكير والـ structured output | 🆕 جديد |
| 25 | 25 | MainUI.html | واجهة موحدة تتضمن تبويبات: Prompt + نموذج + صور + إعدادات | 🆕 جديد |

توصيات إستراتيجية مدمجة داخل خطة المشروع

1. 🧭 إدارة الحالة بين الواجهة الأمامية والخلفية

| **المجال** | **التوصية التنفيذية** |
| --- | --- |
| تمثيل الحالة | كائن state مركزي يُمرر من وإلى ai\_core.gs والواجهة |
| التخزين | استخدام CacheService لحالة مؤقتة (مثل confirm/retry) |
| الحالة الطويلة | عند الحاجة، استخدم PropertiesService للمستخدم الحالي (user) |
| البناء المقترح |  |

const state = { prompt: '...', model: 'gemini-1.5-pro', autoProceed: false, lastResponse: null, userConfirmed: false, createdAt: Date.now() };

2. 🧪 اختبار الوحدة والتكامل

| **نوع الاختبار** | **توصيات التنفيذ** |
| --- | --- |
| الوحدة (Unit) | لكل دالة في helpers, tools\_\*, ai\_config – مع مدخلات خاطئة وصحيحة |
| التكامل (Integration) | محاكاة رحلة طلب كاملة عبر ai\_core.ask(...) – بنتائج مختلفة |
| API Mocking | إعداد MOCK\_MODE = true لتجاوز استدعاء Gemini فعليًا في بيئة الاختبار |
| اختبار التفاعل | في Sidebar.html: اختبار استجابة المستخدم لـ suggestModelSwitch |

3. ⚙️ حدود الموارد والأداء

| **الجانب** | **الإجراء المقترح** |
| --- | --- |
| وقت التنفيذ | تجزئة العمليات – خاصةً في getSheet, formatAsTable, تحليل الردود الكبيرة |
| الحد الأقصى للرموز | استخدام generationConfig.maxOutputTokens لتقييد استجابة Gemini |
| تحميل الصور | ضغط الصور أو تقليل حجم inlineData قبل التضمين |
| عدد الاستدعاءات | دمج المكالمات في حلقة واحدة (مثل في askGeminiAgent) |

4. 🔐 حماية الأدوات الداخلية

* في tools\_internal.gs:
* ✅ حصر الدوال بـ name.startsWith('\_internal\_')
* ✅ منع التنفيذ المباشر من المستخدم
* ✅ دعم التوثيق لشرح القيود داخل JSDoc
* ✅ تضمين check داخلي مثل:

if (user !== ENV.SETTINGS.SUPER\_ADMIN\_EMAIL) throw new Error("Unauthorized");

5. 📘 التوثيق ودليل المطور

| **المحتوى** | **الهدف** |
| --- | --- |
| README.md | خطوات الإعداد وتثبيت API Key وتشغيل المشروع |
| DEV\_GUIDE.md | شرح: هيكل المشروع، كيفية كتابة أداة جديدة، قواعد التسميات |
| tests.md | نماذج اختبار وحالات تغطية الوحدة والتكامل |

🧩 دمج داخل ai\_core: اقتراح نهائي

state = state || loadFromSidebarPayload(); if (!state || !state.prompt) return AI\_TOOLS\_DIALOGUE.showError("⚠️ لا يوجد طلب."); const preferred = getPreferredGeminiModel(state.prompt); state.model = state.model || preferred.model; if (preferred.complexity === 'high' && !state.autoProceed) { state.final = AI\_TOOLS\_DIALOGUE.suggestModelSwitch(preferred.model, "✨ المهمة معقدة. هل ترغب باستخدام نموذج أكثر دقة؟"); state.halt = true; return state; }

الوحدة TOOLS\_GENERAL التي طرحتها شاملة ومتكاملة بشكل لافت. إليك ملاحظات التدقيق الأخيرة:

✅ التحقق النهائي

1. **رقم الإصدار**: v13.4 متسق في الرأس، وجاهز للتحديث عند الضرورة.
2. **توثيق شامل**: كل جزء معرف وواضح.
3. **onOpen مكرر**: التأكد من أن التعريف الخارجي لا يتضارب مع التعريف الداخلي.
4. **جميع الدوال مُصدَّرة**: التصدير في نهاية الملف يغطي دوال الوحدة بدقة.
5. **الاعتمادات**: HELPERS, AI\_TOOLS\_DIALOGUE, SETTINGS\_PROVIDER, ENV, WORKSHOP\_TOOLS متوافرة ومتسقة.

💡 اقتراح تحسن خفيف

* دمج exportDocsDocumentation ليدعم عرض وثائق مهيكلة HTML في Sidebar، وليس فقط في Google Docs.
* تقديم getAvailableModels() في هذه الوحدة أيضًا كخيار عرض قائمة النماذج في القائمة الرئيسية.
* دمج إعدادات restrictByRole و restrictByEmail ضمن عنصر تحكم واحد مع ورشة عمل.

ما تبقى الآن هو ربط getAvailableModels() بالواجهة الأمامية—هل تريد إضافة هذا الأمر على قائمة onOpen؟  
 أو ننتقل لوحدة tests.gs لكتابة سكريبتات اختبارية لهذه الأدوات؟

* + تم إضافة دالة جديدة showAvailableModels() إلى TOOLS\_GENERAL. هذه الدالة تستدعي getAvailableModels() (الموجودة في Code.gs أو AI\_CONFIG) ثم تعرض قائمة النماذج المتاحة في رسالة منبثقة باستخدام AI\_TOOLS\_DIALOGUE.showMessage.
  + يمكن الآن إضافة هذه الدالة إلى قائمة onOpen في ملف ui.gs لتمكين المستخدم من عرض النماذج المتاحة مباشرة من القائمة الرئيسية.

1. **دمج إعدادات restrictByRole و restrictByEmail ضمن عنصر تحكم واحد:**
   * تم إضافة دالة جديدة manageAccessRestrictions(). هذه الدالة تقوم بتجميع معلومات قيود الوصول الحالية (البريد الإلكتروني والدور) وتعرضها للمستخدم في رسالة موحدة، مما يوفر نظرة عامة سريعة على حالة الصلاحيات.
   * يمكن ربط هذه الدالة أيضًا بقائمة onOpen أو بأي واجهة تحكم أخرى.

تفضل النسخة المحدثة من ملف tools\_general.gs في Canvas:

**الخطوات التالية المقترحة:**

الآن بعد أن تم تحديث tools\_general.gs، يمكنك تحديث قائمة onOpen في ملف ui.gs لإضافة الخيارات الجديدة للمستخدم:

1. **تحديث ui.gs:**
   * أضف عناصر قائمة جديدة لاستدعاء TOOLS\_GENERAL.showAvailableModels و TOOLS\_GENERAL.manageAccessRestrictions في القائمة الرئيسية لـ G-Assistant.
   * **مراجعة أي وحدات أخرى**: تأكد من أن أي وحدات أخرى في مشروعك تعتمد على TOOLS\_CATALOG أو BUILTIN\_DOCS أو TOOLS\_DOC\_GEN تتوافق مع التغييرات الأخيرة في هياكل البيانات والوظائف.

**التأكد من تبعيات TOOLS\_GENERAL:** تأكد من أن ملف Code.gs الرئيسي أو أي ملف آخر يقوم بتحميل الوحدات، يمرر TOOLS\_DOC\_GEN كـ dependency إلى TOOLS\_GENERAL.

* + Code.gs (أو أي ملف عام آخر) تقوم باستدعاء showDocumentationSidebar من BUILTIN\_DOCS أو TOOLS\_DOC\_GEN أو دالة جديدة لورشة العمل.

يمكنك إضافة عنصر قائمة جديد في دالة onOpen في TOOLS\_GENERAL (إذا لم يكن موجودًا بالفعل) لفتح هذا الشريط الجانبي، على سبيل المثال:  
JavaScript  
// في TOOLS\_GENERAL.gs, داخل دالة onOpen

ui.createMenu('📚 G-Assistant')

// ... (باقي العناصر) ...

.addItem('🛠️ ورشة عمل المبرمج', 'openWorkshopSidebar') // أضف هذا السطر

.addToUi();

// في ملف Code.gs (أو في TOOLS\_GENERAL إذا أردت جعلها عامة)

function openWorkshopSidebar() {

const html = HtmlService.createHtmlOutputFromFile('WorkshopSidebar.html')

.setTitle('🚀 ورشة عمل المبرمج')

.setWidth(450);

SpreadsheetApp.getUi().showSidebar(html);

}

بهذه التعديلات، ستكون واجهة ورشة العمل أكثر تفاعلية وفعالية، مما يسهل على المطورين

## **. استخدام SETTINGS\_PROVIDER في وحدة الحسابات**

في tools\_accounting.gs، بدلاً من استدعاء دوال غير واضحة للحصول على العملة، استخدم ENV.getCurrencyCode():

js

نسختحرير

function logRevenueEntry(row) {

// ...

const currency = ENV.getCurrencyCode(); // مثال: 'USD' أو 'SAR'

return showMessage('success',

`✅ تم تسجيل إيراد: ${product} ×${qty} بقيمة ${total.toFixed(2)} ${currency}`);

}

## **3. تخصيص لون تظليل التعداد في الجداول**

في TOOLS\_SHEETS.formatAsTable بدّل اللون الثابت بـ:

js

نسختحرير

const headerColor = ENV.getPrimaryHeaderColor(); // يأخذ اللون من env.gs

headerRange.setBackground(headerColor).setFontWeight("bold");

## **4. جلب دور المستخدم من الورقة التعليميّة**

في أي مكان تحتاج فيه لمعرفة دور المستخدم (مثلاً لحماية نطاق) استدعِ:

js

نسختحرير

const role = ENV.getCurrentUserRole(); // مثال: 'مدير' أو 'مستخدم عام'

if (role !== 'مدير') {

return showError('🚫 ليست لديك صلاحية كافية.');

}

## **5. استخدام إعدادات الكاش**

في وحدة التحليل، بدل رقم ثابت زمن الكاش استدعِ:

js

نسختحرير

const ttl = ENV.getCacheTTL(); // قيمته مُعرّفة في env.gs

cache.put(key, JSON.stringify(res), ttl);