## ✅ تحسينات الذكاء الاصطناعي – G-Assistant v12.8

تم معالجة جميع النقاط التقنية المفقودة لتعزيز التكامل مع Gemini API ودعم مزايا متقدمة مثل الصور، التفكير المتسلسل، والإخراج الهيكلي.

# 📘 وحدة HELPERS – الإصدار v16.1

وحدة الدوال المساعدة العامة لمشروع G-Assistant.

تعمل كطبقة دعم مركزية توفر واجهات آمنة لتنفيذ الأدوات، وإدارة الأوراق، وحفظ النسخ الاحتياطية، وتنسيق الجداول.

---

## 🧩 التبعيات

تعتمد على الوحدات التالية:

| الوحدة | الوظيفة |

|--------|----------|

| `ENV` | الوصول إلى الإعدادات والألوان |

| `TOOLS\_CATALOG` | تنفيذ الأدوات المسجلة |

| `AI\_TOOLS\_DIALOGUE` | عرض رسائل للمستخدم |

| `AI\_MEMORY` | تسجيل العمليات وسجل الأداء |

---

## 🚀 الوظائف الرئيسية

### 1. `logOperation(prompt, action, summary)`

يسجل عملية معينة في سجل العمليات عبر وحدة AI\_MEMORY.

```js

HELPERS.logOperation("حساب الربح", "success", { amount: 1200 });

### 📌 جدول التحسينات:

| الميزة | التعديل المُنفذ | الملفات المعنية |

|--------|------------------|------------------|

| \*\*1. دعم الصور (Vision)\*\* | ✅ تم تحويل الصور إلى `inline\_data` (Base64) داخل `parts` تلقائيًا | `ai\_ask.gs` |

| \*\*2. التحكم في "التفكير"\*\* | ✅ تم تمرير `thinkingConfig.thinkingBudget` من `ENV` <br> ✅ دعم `includeThoughts` لعرض خطوات النموذج | `ai\_ask.gs`, `env.gs` |

| \*\*3. السياق الطويل\*\* | ✅ حفظ `systemPrompt` و`lastPrompt` في `CacheService` <br> ✅ استخدام `cachedContent` لتجنب تكرار الإرسال | `ai\_core.gs`, `ai\_memory.gs` |

| \*\*4. الاستجابة المتدفقة\*\* | ✅ تم اعتماد `streamGenerateContent()` من Gemini SDK <br> ✅ عرض تفاعلي للرد في `Sidebar` باستخدام `EventSource` | `ui.gs`, `ai\_ask.gs` |

| \*\*5. إخراج منسق (FunctionCall)\*\* | ✅ دعم `structuredOutput` في `payload` <br> ✅ تحليل وإرسال JSON تلقائي لتنفيذ الأدوات | `ai\_ask.gs`, `tools\_catalog.gs` |

### 🔄 النسخة: v12.8

### 📅 تاريخ التحديث: 2025-07-10

### 🧩 النموذج الافتراضي: `gemini-2.5-flash`

├── env.gs # إعدادات النظام والثوابت

├── helpers.gs # دوال مساعدة وتسجيل العمليات

├── ai\_config.gs # اختيار النموذج وتفضيلاته

├── ai\_context.gs # systemPrompt وتاريخ الجلسة

├── ai\_ask.gs # تكامل مع API وهيكلة المخرجات

├── ai\_memory.gs # ذاكرة الجلسة والذاكرة طويلة المدى

├── ai\_dispatcher.gs # تحليل النية واستدعاء الدوال

├── ai\_core.gs # خط أنابيب التنفيذ وتبديل النماذج

│

├── tools\_catalog.gs # قائمة وتوثيق الأدوات المتاحة

├── tools\_context.gs # إدارة الذاكرة اللحظية

├── tools\_dialogue.gs # رسائل وتأكيدات للمستخدم

├── tools\_sheets.gs # CRUD للجداول، فلترة، تلوين، استخراج روابط

├── tools\_accounting.gs # تسجيل الإيرادات/المصروفات والتقارير المالية

├── tools\_general.gs # نسخ احتياطي، PDF/CSV، إعدادات المستخدم، ورشة العمل

├── tools\_internal.gs # دوال داخلية بعد تأكيد المستخدم

├── builtinDocs.gs # توليد توثيق الأدوات تلقائيًا

├── tools\_doc\_gen.gs # إنشاء مستندات التوثيق برمجيًا

├── workshop.gs # أدوات ورشة العمل (اقتراح ملفات)

├── tests.gs # سكربتات الاختبار الوحدة والتكامل

├── ui.gs # onOpen، قوائم، إطلاق الأشرطة الجانبية

│

├── Sidebar.html # الشريط الجانبي الرئيسي للأوامر والأدوات

├── BuiltinDocsSidebar.html # شريط جانبي للتوثيق

├── PromptGeneratorSidebar.html # توليد أوامر ديناميكية

├── TestResultsSidebar.html # عرض نتائج الاختبارات

└── MainUI.html # واجهة موحدة متعددة التبويبات

## 🌐 المشروع باختصار

G-Assistant هو نظام مساعد ذكي، محاسبي، إداري، وتحليلي، مدمج بعمق داخل Google Sheets. تم بناؤه على \*\*Google Apps Script\*\* ويتكامل مع أحدث قدرات \*\*Gemini API\*\* (مثل فهم الصور، التفكير المتسلسل، والسياق الطويل) لتحويل جداول البيانات من مجرد أداة لإدخال الأرقام إلى \*\*"نظام تشغيل ذكي يستجيب ويتطور معك"\*\*.

#### 🧠 المبدأ الأساسي:

هذا ليس مجرد سكربت، بل هو \*\*مساعد محلي له ذاكرة، وشخصية، وقدرات وظيفية، وقابلية للتطوير الذاتي\*\*. يمكنك التحدث معه بشكل طبيعي، فيقوم بالرد، التحليل، الكتابة، التنفيذ، الاقتراح، وحتى التعليم. إنه بمثابة الجلوس مع محاسب، ومطور، ومحلل بيانات، ومدير تشغيل في آن واحد.

---

## ⚡ المميزات الذكية

| الجانب | ما يميّز مشروعك |

| :--- | :--- |

| \*\*🤖 الذكاء المتقدم\*\* | يفهم السياق، يحلل الصور (الفواتير)، يفكر خطوة بخطوة، ويقترح تبديل النموذج للمهام المعقدة. |

| \*\*📈 المحاسبة الاحترافية\*\* | حساب تلقائي للأرباح، تصنيف المصروفات، وتوليد بيان دخل وتقارير تحليلية قابلة للتخصيص. |

| \*\*🧠 الذاكرة السياقية\*\* | ذاكرة قصيرة المدى (مؤقتة) وذاكرة طويلة المدى (سجل عمليات دائم) لفهم تاريخ وتطور العمل. |

| \*\*🔒 الأمان والصلاحيات\*\* | أدوات تطلب تأكيدًا قبل الحذف، مع نظام حماية للأدوات وحماية للنطاقات بناءً على "دور" المستخدم. |

| \*\*🧩 التطوير الذاتي\*\* | أي أداة ناقصة؟ المساعد يقترحها ويجهز الكود الخاص بها مباشرة في \*\*"ورشة عمل المبرمج"\*\*. |

| \*\*🧾 التوثيق التلقائي\*\* | القدرة على توليد توثيق كامل ومحدث لكل أدوات النظام بضغطة زر. |

| \*\*💼 تعدد الشخصيات\*\* | يتصرف كمحاسب (دقة)، كمبرمج (كفاءة)، كمحلل (رؤية)، وكمصمم (وضوح) حسب سياق الطلب. |

| \*\*🧰 أدوات إدارية\*\* | تصدير PDF، إنشاء نسخة احتياطية آمنة، استعراض معلومات آخر نسخة، وإرسال تقارير بالبريد. |

---

## 🚀 دليل البدء السريع

### 1. المتطلبات

- حساب Google مع وصول إلى Google Sheets و Google Drive.

- مفتاح API من \*\*Google AI Studio\*\* لنموذج Gemini (يفضل Gemini 1.5 Pro للاستفادة من كل الميزات).

### 2. خطوات التثبيت

1. \*\*إنشاء جدول بيانات جديد:\*\* افتح Google Sheets وأنشئ جدول بيانات جديدًا.

2. \*\*فتح محرر السكربت:\*\* من القائمة، اذهب إلى `Extensions > Apps Script`.

3. \*\*إنشاء الملفات:\*\* سيظهر لك ملف افتراضي. قم بإنشاء كل ملفات المشروع (`env.gs`, `ui.gs`, `ai\_core.gs`... إلخ) باستخدام زر `+` في قائمة "الملفات".

4. \*\*نسخ ولصق الأكواد:\*\* انسخ محتوى كل ملف من مصدره والصقه في الملف المطابق له في محرر Apps Script.

5. \*\*ترتيب الملفات (مهم جدًا):\*\* في القائمة اليسرى للمحرر، قم بسحب وإفلات الملفات لتكون بنفس الترتيب المنطقي الذي تم الاتفاق عليه (بحيث يكون `ai\_core.gs` وملفات الاستدعاء في الأسفل).

6. \*\*حفظ المشروع:\*\* انقر على أيقونة الحفظ 💾.

7. \*\*إعداد مفتاح API:\*\*

\* في محرر Apps Script، اذهب إلى "إعدادات المشروع" (أيقونة الترس ⚙️).

\* في قسم "خصائص السكربت" (Script Properties)، انقر على `Edit script properties`.

\* أضف خاصية جديدة:

\* \*\*Property:\*\* `GEMINI\_API\_KEY`

\* \*\*Value:\*\* الصق مفتاح API الخاص بـ Gemini هنا.

\* انقر `Save script properties`.

### 3. أول استخدام

1. \*\*أعد تحميل صفحة Google Sheet.\*\*

2. ستظهر قائمة جديدة باسم \*\*"⚙️ Gemini المساعد الخارق"\*\*.

3. من القائمة، اختر \*\*"🧰 أدوات النظام" > "🧠 إعداد التعليمات والذاكرة"\*\*. هذا سيقوم بإنشاء ورقة `Instructions` التي تمثل "عقل" المساعد.

4. (اختياري) في ورقة `Instructions`، يمكنك وضع اسم النموذج المفضل في الخلية `B1` (مثل `gemini-1.5-pro`) ودورك في الخلية `B2` (مثل `مدير`).

5. أنت الآن جاهز! افتح المساعد من \*\*"💡 فتح المساعد الذكي"\*\* وابدأ العمل.

---

## 🔧 طريقة الاستخدام

- \*\*القوائم المخصصة:\*\* استخدم القوائم للوصول السريع إلى التقارير الرئيسية مثل "حساب الربح الإجمالي" أو "إنشاء نسخة احتياطية".

- \*\*الشريط الجانبي (المساعد الذكي):\*\* تحدث مع المساعد بلغة طبيعية. أمثلة على الأوامر:

- `"سجل مصروف من هذه الفاتورة"` (مع رفع صورة الفاتورة).

- `"أنشئ جدول جديد من قالب 'عملاء'"`

- `"لون كل القيم التي تزيد عن 5000 في العمود C باللون الأصفر"`

- `"أحتاج أداة لحساب الخصومات، جهزها لي في ورشة العمل"`

- \*\*الزر الديناميكي:\*\* في أي ورقة، اكتب اسم دالة (مثل `calculateGrossProfit`) في الخلية `A1`، ثم انقر على زر مخصص قمت بربطه بدالة `runFunctionFromCell` لتنفيذ سريع.

---

## 🏛️ المعمارية والهيكل البرمجي

تم بناء النظام على هيكل معياري متقدم (IIFE) لفصل الاهتمامات وضمان قابلية التوسع.

- \*\*`env.gs`:\*\* لوحة التحكم المركزية للإعدادات.

- \*\*`helpers.gs`:\*\* وحدة الخدمات المساعدة العامة والآمنة.

- \*\*`ai\_\*.gs` (مجموعة ملفات الذكاء الاصطناعي):\*\* وحدات متخصصة لإدارة إعدادات النموذج، بناء السياق، الاتصال بالـ API، وتوزيع المهام.

- \*\*`tools\_\*.gs` (مكتبة الأدوات):\*\* وحدات متخصصة تحتوي على كل القدرات الوظيفية للمساعد (محاسبة، جداول، حوار، توثيق).

- \*\*`tests.gs`:\*\* ملف مخصص لاختبارات الوحدة لضمان استقرار النظام.

# 📘 UI.gs – واجهة المستخدم لـ G-Assistant

الملف `ui.gs` مسؤول عن إنشاء القوائم المخصصة والتفاعل مع Google Sheets عبر الشريط الجانبي والوظائف الديناميكية. الإصدار الحالي \*\*v16.4\*\* يدمج بين الصلابة التنظيمية والمرونة البرمجية باستخدام ذكاء اصطناعي مدمج.

---

## ⚙️ الوظائف العامة

| الدالة | الوصف |

|--------|-------|

| `onOpen()` | تُشغّل تلقائيًا عند فتح الملف، تُحدّث الإعدادات وتبني القوائم |

| `onInstall(e)` | تُنفذ عند تثبيت المشروع لأول مرة، وتستدعي `onOpen` |

| `showSidebar()` | تفتح الشريط الجانبي الرئيسي من ملف `Sidebar.html` |

| `showBuiltinDocsSidebar()` | تفتح شريط توثيق دوال Apps Script، إما عبر وحدة `BUILTIN\_DOCS` أو ملف احتياطي |

---

## 🔁 دوال وسيطة لـ الوظائف الخارجية

الوظيفة `addItem("label", "functionName")` تتطلب دالة معرفة في النطاق العام. لذلك تم تعريف دوال وسيطة تربط تلك العناصر بالوحدات الخارجية.

| الدالة الوسيطة | تستدعي |

|----------------|--------|

| `createBackup()` | `TOOLS\_GENERAL.createBackup()` |

| `generateToolMetadata()` | `TOOLS\_DOC\_GEN.generateToolMetadata()` |

| `showGeneratedDocs()` | `TOOLS\_DOC\_GEN.showGeneratedDocs()` |

| `showToolsDocumentationSidebar()` | `TOOLS\_DOC\_GEN.showToolsDocumentationSidebar()` |

| `exportToolsDocumentationToDoc()` | `TOOLS\_DOC\_GEN.exportToolsDocumentationToDoc()` |

| `summarizeSession()` | `AI\_MEMORY.summarizeSession()` |

| `showAvailableModels()` | `TOOLS\_GENERAL.showAvailableModels()` |

| `manageAccessRestrictions()` | `TOOLS\_GENERAL.manageAccessRestrictions()` |

---

## 🔧 تحديث الإعدادات مع إشعار

- \*\*`refreshEnvSettingsWithNotice()`\*\*

دالة خاصة تستدعي `refreshEnvSettings()` (مفترض أنها موجودة في `env.gs`) وتعرض إشعار نجاح (Toast) أو خطأ.

---

## ✅ مميزات الإصدار v16.4

- تحديث تلقائي لإعدادات المشروع من ورقة التعليمات عبر `onOpen` و `onInstall`

- حماية من كسر القوائم عبر `typeof` لكل دالة خارجية

- قابلية توسعة عالية بفضل `wrapper functions`

- استخدام `ENV.getSidebarFile()` لجلب اسم ملف HTML الديناميكي عند توفره

- عرض Toast للمستخدم يؤكد نجاح التحديث

---

## 📌 ملاحظات

- يجب التأكد من أن دالة `refreshEnvSettings()` موجودة في `env.gs`

- وحدة `BUILTIN\_DOCS` يجب أن تحتوي دالة `showDocumentationSidebar()` لتفعيل السلوك الذكي في التوثيق

- إذا تغيرت أسماء الملفات أو الوحدات، يجب تحديث الدوال الوسيطة بما يتماشى معها

هذا الهيكل يضمن أن النظام ليس فقط قويًا اليوم، بل جاهز للنمو والتطور في المستقبل.