# G-Assistant: The Intelligent Workspace Assistant

\*\*G-Assistant\*\* is a sophisticated, modular assistant designed to operate within the Google Workspace ecosystem, primarily Google Sheets. It leverages the power of Google's Gemini AI to provide advanced financial analysis, operational automation, and powerful developer tools, transforming your spreadsheet into an intelligent command center.

## 🚀 Core Philosophy

This project is built on a foundation of \*\*robustness, scalability, and clarity\*\*. Our goal is to create a professional-grade platform that is not only powerful for end-users (accountants, managers, designers) but also a joy for developers to extend and maintain.

- \*\*Unified Architecture:\*\* A strict, single-namespace structure (`GAssistant`) eliminates global scope pollution and ensures a predictable, organized codebase.

- \*\*Single Source of Truth:\*\* A centralized and cached configuration module (`GAssistant.Config`) manages all settings, making customization and maintenance straightforward.

- \*\*Separation of Concerns:\*\* A clean "Router" (`Code.gs`) handles UI calls, while distinct modules (`AI`, `UI`, `Tools`, `Utils`) manage their specific domains, promoting encapsulation and testability.

- \*\*Developer-Centric:\*\* With built-in code review, documentation generation, and a developer workshop, the project is designed to facilitate its own evolution.

## ✨ Key Features

- \*\*Multi-Faceted Expertise:\*\*

- \*\*Financial Analyst:\*\* Perform complex financial calculations, generate income statements, and detect anomalies in your data.

- \*\*Operations Manager:\*\* Automate data entry, format sheets professionally, and generate daily summaries.

- \*\*Developer's Partner:\*\* Review your Apps Script code for performance, security, and readability. Get instant documentation for any built-in function.

- \*\*Intelligent Interaction:\*\* The assistant understands context, leverages conversation history, and uses the best AI model for the task at hand.

- \*\*Extensible Tooling:\*\* The modular tool catalog allows developers to easily add new capabilities that the AI can immediately discover and use.

- \*\*Secure & Private:\*\* All operations are contained within your Google account. User-specific data is handled with care, using `UserCache` and clear privacy principles outlined in the AI's constitution.

## 📂 Project Structure (The New Architecture)

The refactored project follows a clean, professional structure:

- `initializer.gs`: The first script to run. Sets up the `GAssistant` namespace.

- `Code.gs`: The lean \*\*router\*\* for all `google.script.run` calls from the UI.

- `config.gs`: The \*\*single source of truth\*\* for all project settings.

- `utils.gs`: A consolidated module for logging, sheet manipulation, and other common utilities.

- `ai/`: Contains all AI-related logic.

- `ai\_core.gs`: The central orchestrator for AI requests.

- `ai\_context.gs`: Builds the complete prompt, including the constitution, user role, and sheet context.

- `ai\_memory.gs`: Manages the short-term conversation history.

- `ai\_dispatcher.gs`: Interprets the AI's response (text, tool call, error).

- `ai\_constitution.gs`: Defines the AI's core principles and ethical guidelines.

- `ui/`: For all user interface elements.

- `ui.gs`: Creates menus and manages sidebars.

- `ui\_dialogue.gs`: Creates standardized message objects for the frontend.

- `AssistantSidebar.html`: The main user-facing sidebar.

- `DeveloperSidebar.html`: A tabbed interface for all developer tools.

- `tools/`: All functions that the AI can call.

- `tools\_catalog.gs`: The definitive registry of all available tools and their schemas for the AI.

- `tools\_accounting.gs`, `tools\_sheets.gs`, `tools\_code\_review.gs`, etc.

- `tests/`: For integration and unit tests.

## 🛠️ Installation & Setup

1. \*\*Create Project:\*\* Open a Google Sheet and go to `Extensions > Apps Script`.

2. \*\*Add Files:\*\*

- Delete the default code in `Code.gs`.

- For each file provided in this output, click the `+` icon next to `Files`, choose `Script` or `HTML`, and name it exactly as specified (e.g., `config.gs`, `ai/ai\_core.gs`, `ui/AssistantSidebar.html`). The folder structure is for organization; in Apps Script, you create a file named `ai/ai\_core.gs`.

- Copy and paste the content for each file.

3. \*\*Set API Key:\*\*

- Go to `Project Settings` (⚙️ icon).

- Under `Script Properties`, click `Add script property`.

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

- \*\*Value:\*\* Your Google Cloud API key for Gemini.

4. \*\*Enable APIs:\*\*

- Go to `Services` (+ icon).

- Add and enable the `Google Sheets API`.

5. \*\*Save & Refresh:\*\*

- Click `Save project` (💾 icon).

- Return to your Google Sheet and refresh the page. The `✨ G-Assistant` menu should appear.

## 🤝 Contributing

This project is designed for evolution. To contribute:

1. \*\*Add a Tool:\*\* Create a new function in the relevant `tools/` module (e.g., `GAssistant.Tools.Sheets`).

2. \*\*Register the Tool:\*\* Add its definition (name, description, parameters, and function path) to the `ALL\_TOOL\_DEFINITIONS` array in `tools/tools\_catalog.gs`.

3. \*\*Create a Test:\*\* Add a new test case in `tests/tests.gs` to ensure your tool works as expected.

The AI will automatically discover and be able to use your new tool.

---

Thank you for using and developing G-Assistant!