# Smart Supply Tracker – README

Candidate: SAHIL BHANDARI

Role Applied: Internship – Google Sheets + Apps Script Automation

Task: Track Consumable Medical Supplies and Reorder

## 1. Overview

This document describes the setup, usage, and testing process for the Smart Supply Tracker solution. This solution automates inventory management for clinic consumables using Google Sheets and Google Apps Script. It tracks stock levels, sends automated reorder and confirmation alerts, logs reorder history, and optionally integrates with AI (Google Gemini API) for predictive stock risk analysis.

## 2. Problem Definition & Scope

Healthcare clinics frequently run short on critical consumables due to manual tracking errors and lack of timely alerts. The aim of this automation is to provide:  
• Automatic daily inventory checks.  
• Email alerts when stock falls below a predefined threshold.  
• Accurate reorder history logging.  
• AI-assisted monthly predictions for items likely to run low soon.

## 3. Stakeholders

• Clinic Manager – Receives reorder alerts and manages purchasing.  
• Receptionist – Updates stock usage logs.  
• Nurse – Records consumable usage in real-time.

## 4. Success Metrics

• 100% accuracy in detecting stock below threshold.  
• Reduction in stockouts by at least 80%.  
• Timely alerts delivered daily without manual intervention.  
• Monthly AI predictions reviewed for accuracy above 70% confidence.

## 5. Setup Instructions

### Step 1: Import CSVs

1. Create a new Google Sheet.  
2. Use File → Import → Upload and a popup will appear simply choose “Insert New Sheet” under Import Location and repeat for each csv:  
 • inventory\_master.csv → Inventory Master  
 • usage\_log.csv → Usage Log  
 • reorder\_history.csv → Reorder History  
 • ai\_predictions.csv → AI Predictions  
3. Ensure sheet names exactly match the above.

### Step 2: Add Apps Script

1. Open Extensions → Apps Script.  
2. Delete existing code in Code.gs.  
3. Paste the contents of apps\_script.js.  
4. Save (Ctrl+S) and run the function initialSetup().  
5. Authorize the script when prompted (Click ‘Review Permissions’ and select an email account, preferably the same one in which you are working within the google sheets. If shown that google hasn’t verified this app, simply click ‘advanced’ and click on ‘go to project’. Check ‘Select All’ in project’s access and click continue). 6. Now you will see Execution Started in Execution Log. Simply go to sheets, you will see a prompt, click ok. You will see Execution Completed in Execution Log.  
7. This creates daily and monthly triggers automatically.

### Step 3: Testing

Option A: Simulate consumption  
• Add a row to Usage Log with (Timestamp, Item, QuantityUsed, Notes). Already added demo data, you can edit or add more or simply do nothing.  
• Run applyUsageLog().  
 OR  
Option B: Immediate threshold check  
• Reduce an item's quantity in Inventory Master below its threshold (Already reduced for demo dataset).  
• Run checkInventoryLevels().

**YOU CAN RUN BOTH STEP BY STEP**

### Step 4: AI Integration (Optional)

1. Get a free Google Gemini API key from <https://ai.google.dev/>

IF YOU DON’T KNOW HOW TO GET ONE, FOLLOW BELOW STEPS:

* Search Google Gemini Api Key or go to ( <https://ai.google.dev/> ) on your web browser.
* Sign in with your email account and click on ‘Explore models in Google AI Studio’.
* Click on ‘Get API Key’.
* If you have an API key for Gemini free version then simply copy it and if you don’t then click on create API key. If you are new, then create a new project and your API key will be generated, copy it and if you are an old user, then select existing project generally first one, your API key will be flashed on your screen, simply copy it.

2. In Apps Script → Project Settings → Add Script Property:  
 Property Name: GOOGLE\_API\_KEY  
 Value: Your API Key (which you have copied)  
3. Run runAIPrediction() to generate predictions.  
4. Review 'AI Predictions' sheet output.

## 6. How It Works

• initialSetup(): Creates missing sheets and sets up triggers. • applyUsageLog(): Deducts quantities from inventory based on Usage Log entries.  
• checkInventoryLevels(): Runs daily, sends reorder alerts if stock ≤ threshold.  
• sendReorderAlert(): Sends an email to the Vendor for restock alert and a confirmation email to Owner .  
• logReorder(): Appends reorder details to Reorder History.  
• runAIPrediction(): (Placeholder) Adds AI predictions to AI Predictions sheet.

## 7. Security Notes

• No API keys are stored in code; use Script Properties.  
• Validate inputs to prevent incorrect stock deductions.

## 8. AI Best Practices & Disclosure

This solution uses AI optionally to enhance prediction accuracy. Gemini AI is used to predict items at risk of shortage based on trends. All AI outputs are logged in the AI Predictions sheet for human review before acting. Prompts and outputs should be documented for transparency.

## Appendix

**1. AI Prompts and Output Samples**

**Prompt Used:**

“Given the following inventory and usage log data, predict which items are at highest risk of stockout next month. Return ONLY a JSON array of objects with fields: item, confidence (0-1), notes. No explanations, just JSON”.

Inventory Data: [Inventory Data as JSON]

Usage Data: [Usage Data as JSON]

**Sample AI Output:**

[ {

"item": "Disposable Gloves",

"confidence": 0.85,

"notes": "Trending below reorder level based on recent usage."

},

...]

(Output is logged in the “AI Predictions” sheet)

**2. AI/Automation Integration Summary**

* **Google Apps Script manages:**
  + Daily inventory checks & alerts
  + Usage logging
  + Time-driven triggers (8AM daily check, monthly AI prediction at 6AM)
  + Stock deduction via usage log (applyUsageLog)
  + Automated vendor emails & history logging (sendReorderAlert, logReorder)
* **Gemini AI API:**
  + Predicts high-risk supply items monthly
  + Output reviewed before acting; never auto-orders stock
* **No credentials in code:**
  + API keys must be set as Script Properties

**3. Validation and Manual Correction Steps**

* Inputs (quantity, thresholds) validated/prevented from negative values by script logic
* All AI predictions reviewed by the candidate before decision; errors logged as 'ERROR' in AI Predictions sheet for transparency
* Usage Log status updated on completion for clear audit trail
* Alert Sent flag can be manually reset via utility function for re-testing

**4. Key Automation Logic Overview**

* initialSetup: Ensures all required sheets exist, creates triggers
* applyUsageLog: Deducts quantities and updates usage entries
* checkInventoryLevels: Automatically detects stock below reorder threshold, sends alert emails, logs in history
* runAIPrediction: Calls Gemini API, parses prediction JSON, appends results to AI Predictions sheet; logs errors for debugging
* Email content and structure provided in sample ("Reorder Alert Sent to Vendor")

**5. Disclosure of AI Usage**

* Model/Service: Gemini 1.5 Flash via Google API
* Integration: Monthly trigger for supply risk prediction; result is advisory only
* Documentation: All prompts, outputs, and errors reliably logged in ‘AI Predictions’ CSV and sheet for post-hoc review

**6. Limitations and Known Issues**

* AI outputs are only as accurate as supplied data; edge cases (missing usage entries, spikes) flagged for manual review
* No direct auto-ordering; humans always make final decisions
* API key required for Gemini calls—never present in code for security; reviewer must insert their own
* Demo data provided; additional sample entries can be manually added for extended testing
* Future improvement: smarter validation, multi-vendor support, failover for API downtime, enhanced reporting

**7. API/Key Placeholders & Reviewer Setup Notes**

* Reviewer must obtain their own Gemini API key (see Section 4, Step 4 in README)
* Key is set in Apps Script Properties via UI, never directly in script
* Detailed setup and function instructions in README for easy review and execution

**8. Extra Credit & Transparency**

* Error-handling logic for AI output parsing and JSON validation
* All actions logged—Inventory Master, Usage Log, Reorder History, AI Predictions
* Commentary in Apps Script for maintainability; utility function for manual alert reset
* Security notes included to prevent inadvertent credential exposure