On-the-Go AppSuite Documentation

Russell Nimmo – Assisted by Google Gemini & Grok - **November 2025**

Table of Contents

Section 1: How it Works

How the On-the-Go AppSuite Works

This system is a simple and cost-effective safety net built using web technologies. It's made of two apps and a smart Google Sheet that ties them together.

1. **The Worker App (PWA):** This is the app for your worker's smartphone. When they're about to start a visit, they open the app, pick a location, and set a timer (e.g., 45 minutes). This action sends their details and location to the backend. The app also handles in-app safety features like the Panic Button, Check-ins, and Duress PIN.
    - **Simple Mode:** When the worker is safe, they tap "Depart," and the timer is cleared.
    - **Advanced Mode:** When the worker is safe, they tap "Depart & File Report." A checklist appears (either standard or custom) for them to fill out before the visit is logged. It also features a "Quick Stat" button for non-safety-related data entry.
2. **The Google Apps Script Backend (The Brain):** This is a Google Sheet with a script attached. When the Worker App sends an update, this script catches the data and writes it as a new row in the spreadsheet. This script also runs on an automatic 5-minute trigger. If it checks the sheet and finds a worker is overdue or has missed a check-in, it automatically sends the alert emails and SMS messages to your emergency contacts.
3. **The Monitoring Dashboard:** This dashboard is for your real-time monitor. Every 15 seconds, it securely fetches all the current data from the Google Sheet and displays it in a clean, prioritized list. This allows the monitor to see who is on-site, who is in an alert state, and where they are, all in real-time. Entries marked "DATA_ENTRY_ONLY" (from a Quick Stat) are ignored by the safety monitor.

Section 2: Deployment Administrator Guide

Welcome! This guide explains how to use the online Setup Tool to create your customized On-the-Go AppSuite and deploy them using GitHub Pages.

Goal: To generate and host your own private versions of the Worker App and Monitoring Dashboard.

Estimated Time:

- Reading: 15-20 minutes
- Set-up: 20-60 minutes. This is a one-time process.

Part 1: Use the Online Setup Tool

To access the On-the-Go AppSuite Setup Tool website, visit: **[Your Hosted setup.html URL]**

1. You should see the "On-the-Go AppSuite - Setup Tool".
2. Follow the steps presented in the tool (Slightly expanded below):

Step 1: Welcome

- Read the introduction. (Note... If you are reading this file, then you already have the Full Documentation and there is no need to download it again.)
- Click **"Start Setup"**.

Step 2: Sheet Setup

- **Create Sheet:** Do you already have a Google account? If not, you will need to sign up for a free Google Account first. (For best data protection it might be wise to create one specifically for this purpose.) Once you know the account that you plan to use, sign into the account and then proceed to the next step.
- Click the **sheets.new** link to open a blank Google Sheet. Give it a name (e.g., "My Team OTG Data").
- **Rename Visits Sheet:** The first sheet tab (default named "Sheet1") will be your safety log. **Rename this tab to Visits**.
- **Copy Headers:** Click the **"Copy Headers"** button in the setup tool. (This button is smart: it will show 19 headers for the "Simple" app and 21 headers if you check "Advanced Reporting" in the next step).
- **Paste Headers:** In your Google Sheet, click cell **A1** and **Paste**. The headers (Date, Worker Name, ..., Battery Level) should fill row 1 automatically.
- **Protect the Header Row:** Optionally, you may like to protect the headers from accidental deletion. (Click on the row label **1** to highlight the row, then right-click and select View more row options > Protect Range. Enter a description, then click on Set Permissions and select to have it warn you.)
- **Create Checklists Sheet (Advanced Reporting Only):** If you plan to enable Advanced Reporting, click the **+** icon at the bottom of the sheet to add a new tab. **Rename this new tab to Checklists** (must be exact).
    - In this new **Checklists** sheet, click cell **A1** and type **Company Name**.
    - In cell **B1**, type **Question 1**. In **C1**, type **Question 2**, and so on.
    - **Add (Standard) Row:** In cell **A2**, you **must** type **(Standard)** (including the parentheses). This is the "magic key" the system uses as the default fallback form.
    - In cells B2, C2, etc., add your questions.

    - **Syntax for Questions:**
        - **Checkbox:** Client goals reviewed
        - **Section Header:** "# Section 1: Safety" (starts with #)
        - **Text Note Box:** "% Strengths Noted" (starts with %)
        - **Number Box:** "$ Kilometres Travelled" (starts with $)
- Click **"Next Step"** in the setup tool.

Step 3: Configure & Deploy Script

- **Enter Organisation Name:** (e.g., "ACME Corp"). This will brand your apps.
- **Enter Secret Key:** In the setup tool, type a **strong, private password**. **Remember this key!** Write it down somewhere safe as it will be needed by the Monitoring Apps.
- **Choose App Type:**
    - **Simple (Default):** Leaves the box unchecked for a safety-only app. The "Depart" button will simply end the visit.
    - **Advanced (Checkbox):** Check this box to enable all visit reporting features. This will enable the "Depart & File Report" workflow in the Worker App and activate the "Gemini API Key" field below.
- **(Optional) Gemini API Key:** If you enabled Advanced Reporting, you can paste your API key here. (See **Section 5** of this guide for instructions on how to get your key).
- **Copy Script:** Click the **"Copy Script"** button. This copies the correct backend code (simple or advanced) with your keys and Organisation Name injected.
- **Paste Script:** In your Google Sheet, click **Extensions > Apps Script**. Delete any existing code in Code.gs and paste the copied code.
- **Deploy:** Click **Deploy > New deployment**. Select type **Web app**, Execute as **Me**, Who has access **Anyone**. Click **Deploy**.
- **Authorize:** Allow the script access to your Google Account (click "Advanced", "Go to...", "Allow"). You may need to do this twice if you are enabling advanced features (for Drive and Docs permissions).
- **Copy URL:** After deployment, **copy the Web app URL**. (You need it in the next step, but also paste it somewhere else secure.)

- **Set Trigger:** In Apps Script, click "Triggers" (alarm clock icon), click **Add Trigger**, set it to run **checkOverdueWorkers** on a **Time-driven** trigger, **Minutes timer**, **Every 5 minutes**. Click **Save**.
- **IMPORTANT - How to Update Your Script:** If you ever need to edit the Code.gs file later (e.g., to add an API key), you **must re-deploy** it. Click **Deploy > Manage deployments**, click the **pencil (Edit)** icon, select **"New version"** from the "Version" dropdown, and click **Deploy**.
- Click **"Next Step"** in the setup tool.

Step 4: Configure Apps

- **Web App URL:** Paste the **Web app URL** you just copied. This is the "brain" that your Worker and Monitor apps will talk to.
- **Configure:** Set the First Alert Time, **Alert Escalation Time**, Check-in options, and optional Logo URL.
- **Test:** Click **"Test Connection & Proceed"**. If successful, you'll move to the last step. (If not, re-check your Web App URL and Secret Key).

Step 5: Download

- Click the **"Generate & Download App Package (.zip)"** button.
- Save the **OTG_AppSuite_Package.zip** file to your computer.

Step 6: Finish

- You are now ready to host your apps.

Part 2: Host Your Generated Apps using GitHub Pages

Now you'll upload the apps from the .zip file to your own GitHub repository to make them live.

1. **Prepare Your Files:**
    - **Unzip** the OTG_AppSuite_Package.zip file you downloaded.
    - Inside, find the **WorkerApp** folder and the **MonitorApp** folder.
2. **Create** a GitHub Account (If you **don't have one):**
    - Go to https://github.com/ and create a free account.
3. **Create a New Repository:**
    - Log in to GitHub. Click **"+"** > **"New repository"**.
    - Repository name: Choose a name, e.g., **my-otg-apps**.

   - Select **"Public"** (this is required for free hosting).
   - Click **"Create repository"**.

4. **Upload App Folders:**
   - On your new repository page, click **"Add file"** > **"Upload files"**.
   - Drag both the **WorkerApp** folder AND the **MonitorApp** folder onto the upload area.
   - Click **"Commit changes"**.

5. **Enable GitHub Pages:**
   - Go to your repository's **"Settings"** tab.
   - Click **"Pages"** in the left sidebar.
   - Under "Build and deployment", set "Source" to **"Deploy** from a **branch"**.
   - Under "Branch", select **main**, folder **/root**, and click **"Save"**.
   - **Wait:** GitHub will provide a public URL (e.g., https://your-username.github.io/my-otg-apps/). It might take a minute or two to become active.

6. **Get Your App URLs:**
   - **Worker App URL:** [Your GitHub Pages URL] + /WorkerApp/
   - **Monitor App URL:** [Your GitHub Pages URL] + /MonitorApp/
   - Save these two URLs.

Part 3: Distribute to Your Team

For Your Safety Monitor:

- Send them the **Monitoring App URL**.
- **IMPORTANT:** On their first visit, they must enter the **Web app URL** (from Step 3) and the **Secret Key** (from Step 3). This is a one-time setup.

For Your Workers:

- Send them the **Worker App URL**.
- **Instruct them to:**
  1. Open the link on their smartphone.
  2. Use their browser's menu to **“Add to Home Screen”** or **"Install app"**.

3. Open the app from their new home screen icon.
4. Go to **Settings** (gear icon) and fill in **all fields** (Name, Phone, Contacts, PINs). The URL is pre-filled.
5. (If using Advanced Reporting) Go to the **Main Screen** and add their visit locations, assigning the correct **Company Name** and **Template Name**.
6. Tap **"Save Settings"**.

Setup Complete! Your system is operational.

Section 3: (Optional) Create an Installer for the Monitoring App

This guide will walk you through turning the Monitoring Dashboard URL into a standalone desktop application (like an .exe file on Windows). This is highly recommended as it prevents a monitor from accidentally closing the browser tab.

We will use a free, trusted tool called Nativefier.

1. Install the Necessary Tools (Node.js)

1. **Go to the Node.js Website:** https://nodejs.org/
2. **Download the Installer:** Click the button for **"LTS"** (Long Term Support).
3. **Run the Installer:** Find the downloaded file and double-click it. Accept all default settings.

2. Open Your Command Line Tool

- **On Windows:** Press the **Windows Key**, type cmd, and click on **"Command Prompt"**.
- **On macOS:** Click **Spotlight** (magnifying glass), type Terminal, and click on **"Terminal.app"**.

3. Install Nativefier

- In the command-line window, type the following and press **Enter**: npm install -g nativefier
- Wait for it to finish. You only need to do this once, ever.

4. Build Your Desktop App

1. **Go to Your Desktop:** In your command-line window, type the following and press **Enter**: cd Desktop
2. **Run the Nativefier Command:**
   - Get your **Monitoring App URL** from Part 2.6.

  - Carefully type the following, **replacing the placeholder URL** with your actual Monitor App URL. nativefier --name "OTG Monitor" --internal-urls ".*" "https://your-username.github.io/my-otg-apps/MonitorApp/"
  - Press **Enter**. Nativefier will take a few minutes to build your app.

3. **Find Your App:**
   - Look on your computer's **Desktop**.
   - You will find a new folder (e.g., OTG Monitor-win32-x64).
   - Inside this folder is your new application (OTG Monitor.exe or OTG Monitor.app).

5. Share the App with Your Monitor

- You must send the **entire folder** Nativefier created.
- **Right-click** on the folder and select **"Send to" > "Compressed (zipped) folder"**.
- You can now send this single .zip file to your monitor via email.
- **Instructions for Monitor:** Tell them to unzip the file, open the folder, and run the OTG Monitor application.

6. (Recommended) Make the Monitor App Start Automatically

- **On Windows:**
  1. Move the OTG Monitor-win32-x64 folder somewhere permanent (like C:\Program Files).
  2. Inside that folder, **right-click** OTG Monitor.exe and select **"Create shortcut"**.
  3. Press **Windows Key + R**, type shell:startup, and press **Enter**.
  4. Drag your new **shortcut** into the "Startup" folder.
- **On macOS:**
  1. Move the OTG Monitor.app file to your **Applications** folder.
  2. Open **System Settings** > **"General"** > **"Login Items"**.
  3. Click the **plus icon (+)** and add the OTG Monitor.app from your Applications folder.

Section 4: (Optional) Spreadsheet Administrator Guide

This guide is for the person managing the Google Sheet database. This role involves setting up report templates, running monthly reports, and basic data maintenance.

4.1 How to Set Up Monthly Reports (Spreadsheet Tabs)

This function generates a separate, clean spreadsheet *tab* for each company's monthly data.

1. **Manually Create Tabs:** In your Google Sheet, create a new tab named **Master Report**.
2. **Go to the Master Report tab.**
3. In cell **B1**, type the month you want to report on (e.g., 2025-10).
4. **Create a Button (One-time setup):**
   - Go to **Insert > Drawing**.
   - Create a button shape and type "Generate Spreadsheet Reports" on it.
   - Click **"Save and Close"**.
   - Drag the button where you want it. Click it once, click the **three-dot menu** in its corner, and select **"Assign script"**.
   - In the box, type: generateMasterMonthlyReport
   - Click **OK**.
5. **Run the Report:** Click the button. The script will run and create new tabs (e.g., Report - Smith & Co.) with all the data for that month, including a new "Numeric Totals" section.

4.2 How to Set Up PDF Reports (Google Doc Mail Merge)

This function generates professional, multi-page **PDFs** for each company and saves them to a Google Drive folder.

1. **Create your Google Doc Templates:**
   - Create one or more Google Docs to act as your templates.
   - Design them with your logo, text, and placeholder "tags" like {{CompanyName}}, {{TotalVisits}}, {{TotalHours}}, {{ChecklistTable}}, {{NumericTotalsTable}}, and {{NotesTable}}.
   - For each template, open it and copy its **ID** from the URL (the long string between /d/ and /edit).
2. **Create your Output Folder:**

    - In Google Drive, create a folder (e.g., "Monthly Reports - Generated").
    - Open the folder and copy its **ID** from the URL (the string after /folders/).
3. **Configure the Script:**
    - Go to **Extensions > Apps Script**.
    - At the top of the Code.gs file, paste your IDs into these lines:
        - var DEFAULT_REPORT_TEMPLATE_ID = "YOUR_DEFAULT_TEMPLATE_ID_HERE";
        - var PDF_OUTPUT_FOLDER_ID = "YOUR_OUTPUT_FOLDER_ID_HERE";
    - Save and **Re-deploy** (Deploy > Manage deployments > Edit > New version).
4. **Link Templates to Companies:**
    - Go to your Checklists sheet.
    - Find an empty column (e.g., G1) and type the header: Report Template ID
    - In that column, paste the Google Doc Template ID for each company. **You must add one for the (Standard) row.** This is the fallback template.
5. **Create the Button:**
    - Go to the Master Report tab.
    - Insert a new drawing (button) and label it "Generate All PDF Reports".
    - Assign it the script name: generateAllPdfReports
6. **Run the Report:** Click the button. The script will create all the PDFs in your Google Drive folder.

4.3 Setting up Longitudinal (Year-over-Year) Reports

This feature appends monthly totals to a separate, dedicated spreadsheet for each company, allowing you to build graphs that track trends over time.

A) One-Time Setup per Company:

1. Go to your Checklists sheet.
2. Find an empty column (e.g., H1) and type the header: Longitudinal Report Sheet ID
3. Go to **Extensions > Apps Script**.

4. From the function dropdown at the top, select createLongitudinalWorkbook.
5. Click **Run**.
6. A prompt will appear. Enter the *exact* Company Name (e.g., Smith & Co.) and click OK.
7. The script will create a new Google Sheet file ("Longitudinal Report - Smith & Co."), create all the correct headers (including your new $ numeric fields), and **automatically paste the new Sheet's ID** into the Longitudinal Report Sheet ID column for you.
8. Repeat this process for every company you want to track.

B) Running the Monthly Append:

1. Go to the Master Report tab.
2. Insert a new drawing (button) and label it "Append Longitudinal Data".
3. Assign it the script name: runAllLongitudinalReports
4. Click **OK**.
5. Now, when you have a month in cell B1 and click this button, the script will open every linked workbook and append that month's summary as a new row.

4.4 Database Maintenance (Trimming the Visits Sheet)

Your Visits sheet will grow over time. To keep the sheet fast and responsive, you should periodically archive old data.

Warning: This action is permanent. Always make a backup first (File > Make a copy).

1. Go to the **Visits** sheet.
2. Click the filter icon on **Column A (Date)** and sort it A→Z (oldest to newest).
3. Select the rows you want to archive. **CRITICAL: Do NOT delete Row 1 (the headers).**
4. **Right-click** on the highlighted row numbers.
5. Select **Delete rows [2 - XX]**.

Section 5: (Optional) How to Get a Gemini API Key

This step is only required if you enabled "Advanced Reporting" and want to use the AI-powered spelling and grammar correction for your PDF reports.

This process involves three stages:

1. **Get the Key** from Google AI Studio.
2. **Enable the API** in the Google Cloud Console.
3. **Enable Billing** in the Google Cloud Console (this is required by Google, but you won't be charged for this small usage).

5.1 Get the API Key

1. Go to the Google AI Studio website: **https://aistudio.google.com/**
2. Sign in with the **same Google Account** you used for your Google Sheet.
3. On the left-hand menu, click on **"API keys"**.
4. Click the **"Create API key"** button. You may be asked to create a new "Google Cloud project." This is normal. Give it a name (e.g., "My-OTG-App-Project") and click "Create".
5. A new API key (a long string of letters and numbers) will be generated for you. **Copy this key.**
6. Paste this key into the **Gemini API Key** field in the **Setup Tool (Step 3)** *before* you copy the script.
   - **If you already deployed your script:** You can paste this key directly into your Code.gs file (on line 8) and then **re-deploy** your script (see the "How to Update Your Script" note in Step 3).

5.2 Enable the API

Creating the key *does not* automatically turn it on.

1. Go back to the **"API keys"** page in Google AI Studio.
2. You will see your new key listed. Click on the **Google Cloud project name** written next to it.
3. This will open the Google Cloud Console.
4. In the search bar at the top, type **Generative Language API** and select it from the results.
5. A new page will load. If you see a blue **"ENABLE"** button, click it.

5.3 Enable Billing (Required)

Google requires a billing account to be linked to the project to use the API, even for free-tier usage.

1. In the Google Cloud Console, click the "hamburger" menu (☰) in the top-left corner.
2. Select **"Billing"**.
3. Look at the project you just created. If it says "This project has no billing account," you must link one.
4. Click **"Link a billing account"** and follow the prompts to create a new billing profile.
5. You will **not** be charged for this app's usage. The gemini-2.5-flash-preview-09-2025 model has a very large free tier, and your monthly report generation will not come close to exceeding it. This is purely for identity verification.

Section 6: Worker App User Guide

1. Introduction

Welcome! This guide explains how to install, set up, and use the On-the-Go AppSuite.

The purpose of this app is to provide an extra layer of safety when you are working alone. If your organization has enabled "Advanced Reporting," it also helps you file your visit reports.

2. Installation (Adding the App to Your Phone)

This app is a Progressive Web App (PWA). You must install it from the web link your administrator provided. For the app to work reliably, you **must "Add to Home Screen"**.

On iPhone (Safari)

1. Open the app link in your **Safari** browser.
2. Tap the **Share** icon (a square with an arrow pointing up).
3. Scroll down the menu and tap on **"Add to Home Screen"**.
4. Confirm the name and tap **"Add"**.
5. The app icon will now be on your phone's home screen. **Please open the app from this icon** from now on.

On Android (Chrome)

1. Open the app link in your **Chrome** browser.
2. Tap the **three-dot menu** icon (usually in the top-right corner).
3. Tap on **"Install app"** (or **"Add to Home Screen"**).
4. Confirm the installation.

5. The app icon will now be on your phone's home screen. **Please open the app from this icon** from now on.

3. First-Time Setup (Crucial)

Before you can use the app for the first time, you **must** complete your settings.

1. Open the app from the new icon on your home screen.
2. Tap the **Settings icon** (the gear symbol in the top-right corner).
3. Fill in **ALL** the fields:
   - **Your Name:** Your full name.
   - **Your Phone Number:** Please use the full international format, including your country code (e.g., +64 21 123 4567).
   - **Emergency Contact (Name, Phone, Email):** The *first person* to be contacted in an alert.
   - **Escalation Contact (Name, Phone, Email):** The *second person* to be contacted if you are overdue for a long time.
   - **Google Sheet Web App URL:** This should already be pre-filled for you.
4. **Set Your PINs:** You must set two different 4-digit numbers.
   - **Normal 4-Digit PIN:** This is your "all clear" PIN. You will use this to confirm you are safe or to extend your time.
   - **Duress 4-Digit PIN:** This is your **silent alarm**. You *only* use this PIN if you are being forced by someone to cancel an alert or extend your time. (See Section 6 for details).
5. Tap **"Save Settings"**. The app is now ready to use.

4. Managing Your Locations

Before your first visit, you can save the locations you visit frequently.

- **To Add a Location:**
  1. On the main screen, tap the **"Add New Location"** button.
  2. Enter a **Location Name** (e.g., "Smith Residence", "Office Carpark").
  3. Enter the **Location Address**. (You can tap "Use Current Location" if you are there).
  4. **(Advanced Reporting Only)**

        - **Company Name:** Enter the company name for this location (e.g., "Smith & Co.").
        - **Template Name (Optional):** If this visit requires a special report form, type its name here. If you leave this blank, the app will use the (Standard) form.
    5. Tap **"Save"**.

- **To Edit a Location:** Tap the small **pencil icon** next to any location in the list.
- **"Travelling" Location:** This is a default location. Select this when you are driving or moving between sites. When "Travelling" is selected, the app will try to send a GPS update to your monitor every 15 minutes (see limitations in Section 7).

5. Using the App for a Visit

This is the main day-to-day workflow.

1. **Select Location:** On the main screen, tap the location you are visiting.
2. **Set Duration:** Use the slider to set how long you expect to be.
3. **Start Timer:** Press and **HOLD** the green **"Start"** button for **1.5 seconds**.
4. The screen will change to the **Locked Screen**, and your timer will begin. The app sends an ON SITE status to the monitor.

(Advanced Reporting Only) Filing a "Quick Stat" Use this for logging data *without* starting a safety timer (e.g., a phone call or admin task).

1. Select a **Location** from the main screen (do not set a duration).
2. Tap the **"Quick Stat"** button.
3. The report form for that location will appear immediately.
4. Fill out the form and click **"Submit Report"**. This logs your data with a status of DATA_ENTRY_ONLY.

6. Safety & Alert Features

6.1. Panic Button (SOS)

In a real emergency (e.g., injury, threat, accident), **TRIPLE-TAP the red "SOS" button** on the Locked Screen. This immediately sends a high-priority EMERGENCY - PANIC BUTTON alert to your monitor.

6.2. Extending Your Time

If you are running late:

1. Press and **HOLD** the **"Extend 10 Mins"** button for **1 second**.
2. A keypad will appear. Enter your **Normal 4-Digit PIN**.
3. The timer will add 10 minutes, and a note is sent to the monitor.

6.3. Departing Safely (Ending Your Visit)

When you are finished and are leaving the site safely:

1. Press and **HOLD** the red **"DEPART & FILE REPORT"** button (or "DEPART" button in Simple Mode) for **1.5 seconds**.
2. **If you are in Advanced Mode (and not at "Travelling"):** The **"Visit Report"** modal will appear.
   - Fill out the checklists, note fields, and any number fields (e.g., $ KMs).
   - Click **"Submit Report"**.
3. **If you are in Simple Mode (or at "Travelling"):** The app will send a DEPARTED status to the monitor and return you to the main screen. Your visit is now complete.

6.4. "Are You OK?" Check-ins (If Enabled)

- **What happens:** At a set interval (e.g., every 30 minutes), a box will pop up asking "Are you OK?".
- **Escalating Alert:** The app will beep and vibrate, getting louder and more frequent over 2 minutes.
- **Your Action:** You have **2 minutes** to tap the green **"I am OK"** button.
- **Missed Check-in:** If you do not tap "I am OK" within 2 minutes, the app sends a MISSED_CHECKIN alert.

6.5. Cancelling an Alert

If your timer is overdue, the "ALERT ACTIVE" screen will appear.

1. Press and **HOLD** the green **"I AM SAFE"** button for **1 second**.
2. Enter your **Normal 4-Digit PIN**.
3. This will send a SAFE - MANUALLY CLEARED status and (in Advanced Mode) open the **Visit Report** modal for you to complete.

6.6. THE DURESS PIN (Your Silent Alarm)

Use this PIN if you are being forced by a hostile person to silence an alarm or extend your time.

If you are prompted for your PIN (either for "I AM SAFE" or "Extend 10 Mins") and you are in danger:

1. Enter your **DURESS 4-Digit PIN** instead of your Normal PIN.
2. **What the app will do:** The app will *look like it worked*. The timer will extend or the visit will end.
3. **What it *really* does:** The app silently sends a high-priority DURESS_CODE_ACTIVATED alert to your monitor.

7. CRITICAL APP LIMITATIONS (Must Read)

This is a web-based app and has limitations you **must** understand.

7.1. GPS and Background Freezing

For GPS tracking to work reliably, the app **MUST be open and active on your screen.**

- If you **lock your phone** or **switch to another app**, the operating system (especially iOS) will **FREEZE** the app to save battery.
- **When frozen, the app CANNOT:**
    - Send its 15-minute "Travelling" GPS updates.
    - Run its internal check-in timer.

7.2. Alerts and "Waking Up"

- **Server-Side Alerts:** Even if your app is frozen, the **backend server** (Google Sheet) knows when your visit is supposed to end. If you become overdue, the server **will still send automated email alerts** and SMS messages to your contacts.
- **"Waking Up" the App:** If an alert is triggered *while the app is frozen*, the app **WILL** try to send its latest GPS location *as soon as it becomes active again*.

7.3. Network Connection

The app requires an internet connection (Mobile Data or Wi-Fi) to send all **safety alerts**.

- **(Advanced Mode) Offline Reporting is OK:** Your visit reports *can* be saved offline and will sync later.
- **(All Modes) Safety Alerts are NOT Offline:** A PANIC or DURESS alert cannot be sent if you have no network connection.

Section 7: Monitor App User Guide

1. Introduction

Welcome to the On-the-Go Monitoring Dashboard. Your role as a safety monitor is the most critical part of this system.

2. First-Time Login (Crucial)

The first time you open the **Monitoring App URL**, you will see a setup screen.

1. **Paste the Google Sheet Web App URL:** Your administrator will give you this.
2. **Enter the Secret Key:** Your administrator will give you this.
3. Click **"Start Monitoring"**. The app will save these details in your browser. You will not have to log in again unless you clear your browser data or use the "Reset" button.

3. Understanding the Dashboard

The dashboard is designed to give you all critical information at a glance.

- **Header:** Shows your connection status, notification status, and the time of the last successful data refresh (it checks every 15 seconds).
- **Session Event Log:** A collapsible log that shows a timestamped history of events *during your current viewing session*.
- **The Worker List:** Shows a "card" for every worker who is **currently active**. Cards are sorted by alert priority (alarms at the top).

4. Understanding a Worker Card

- **Worker Name:** The name of the worker.
- **Battery Level (%):** Shows the worker's phone battery percentage.
- **Location Name:** The location the worker selected (e.g., "Smith Residence", "Travelling").
- **Status & Due Time:** Shows the current Alarm Status and their Anticipated Departure Time.
- **View Location (Alerts Only):** This link appears when a worker is in an alert state. Clicking it opens Google Maps to their last known location.
- **Call Worker Button:** A blue "Call" button to call the worker's phone.
- **Manually Resolve Alert (Alerts Only):** A green button to clear an alert.

5. Alert Statuses & Responses

The card's color is your most important indicator.

- **GRAY - ON SITE:**

    - The worker is safe. No action is needed.
- **AMBER BORDER (Still says ON SITE):**
    - **Pre-Alert / Overdue.** The worker's timer has run out, but automated email/SMS alerts have not started yet.
    - **Action:** This is a good time to proactively try calling the worker.
- **ORANGE/YELLOW - ALERT SENT / MISSED_CHECKIN / EMAIL_1_SENT...**
    - **Active Alert.** The worker is officially overdue or has missed a check-in. The system is sending automated email/SMS alerts.
    - **Action:** Follow your organization's safety procedures. Call the worker immediately.
- **RED (FLASHING) - EMERGENCY - PANIC BUTTON:**
    - **Critical Alert!** The worker has triple-tapped the **SOS** button.
    - **Action:** This is your highest priority. Treat it as a real emergency. Call the worker and/or emergency services immediately.
- **PURPLE (FLASHING) - DURESS_CODE_ACTIVATED:**
    - **Critical Alert!** The worker has entered their **Duress PIN**. They may be under threat.
    - **Action:** Treat this as a real emergency. **Do NOT** call the worker. Follow your organization's duress procedure (e.g., notify emergency services or security immediately).

5.1. The Alarm Overlay (Loud Alarm)

When a **new** critical alert (PANIC, DURESS, MISSED_CHECKIN, or ALERT SENT) first appears, a **loud, repeating alarm sound** will play. To silence the alarm, you **must** click the **"Acknowledge Alert"** button. This silences the sound but does *not* clear the alert.

5.2. Desktop Notifications (If Enabled)

If you have enabled desktop notifications, you will also see a notification pop up from your browser, even if it is minimized.

6. Your Actions as a Monitor

6.1. Manually Resolving an Alert

When you have made contact with a worker and confirmed they are safe:

1. On the worker's alert card, click the green **"Manually Resolve Alert"** button.

2. A confirmation box will appear. You **must** type the worker's name exactly to confirm.

3. Click **"Confirm Resolve"**. The worker's card will disappear.

6.2. Enabling Desktop Notifications

1. In the header, if you see a "Disabled" status, click the **Bell Icon**.

2. Your browser will pop up a box asking for permission. Click **"Allow"**.

3. The status will change to "Enabled", and you will receive a test notification.

6.3. IMPORTANT: Browser vs. Desktop App (Reliability)

This Monitoring Dashboard is a website. If you **accidentally close the browser tab**, all monitoring **will stop**. To prevent this, it is **highly recommended** that you use the optional **Desktop Installer version** of this app (if provided by your administrator, see Section 3 of this guide).

Section 8: Troubleshooting Guide (For Admins)

Here are solutions to the most common problems.

- **Problem:** The "Start Setup" button on the setup tool website doesn't work.
  - **Cause:** The tool can't load its template files.
  - **Solution:** Make sure you are running setup.html from the **live GitHub Pages URL** (e.g., https://...github.io/...), not by double-clicking the file from your computer (e.g., file:///...).
- **Problem:** My Worker App (Advanced) won't install / The "Travelling" location is missing / The "Settings" button doesn't work.
  - **Cause:** A JavaScript error on load, usually because the (Standard) checklist is missing from your Checklists sheet.
  - **Solution:**
    1. Go to your Checklists sheet.
    2. Make sure you have created a row with the Company Name **(Standard)** (with parentheses) and added at least one question to it.
    3. On the worker's phone, go to **Settings** and click **"Clear Cached Forms & Data"**.
    4. Restart the app.

- **Problem:** My Custom Form shows "#Header" or "%Note" with a checkbox.
  - **Cause:** This means the backend script (Code.gs) is running an old version of the code *before* we added this feature, or you have a typo in your Checklists sheet.
  - **Solution:**
    1. First, check your Checklists sheet. Make sure you add a **space** after the #, %, or $ (e.g., "# Section 1", "% Notes", "$ KMs").
    2. If that is correct, re-deploy your script. Go to Extensions > Apps Script, click Deploy > Manage deployments, edit your deployment, select **"New version"**, and click **Deploy**.
    3. After fixing, clear the cache on the Worker App (see above).
- **Problem:** The Monitor App shows "Cannot Connect" or a 404 error.
  - **Cause:** The Google Sheet URL or Secret Key is wrong.
  - **Solution:**
    1. Re-copy the **Web app URL** (not the Deployment ID) from Manage deployments in your Apps Script.
    2. Go to the Monitor App, click the **Reset/Gear icon** in the header.
    3. Re-paste the correct URL and re-type your Secret Key.
- **Problem:** My generate...Report script fails or reports are blank.
  - **Cause:** Mismatch in names or date format.
  - **Solution 1:** Make sure the month in the Master Report tab (cell B1) is in YYYY-MM format (e.g., 2025-10).
  - **Solution 2:** Make sure the Company Name in the Visits sheet (entered by workers) *exactly* matches the Company Name in your Checklists sheet (it is case-sensitive).
- **Problem:** My AI Note Correction isn't working (notes are uncorrected).
  - **Cause:** The API Key, Billing, or API is not set up correctly.
  - **Solution:**
    1. Go to **Extensions > Apps Script** and open your **Code.gs** file.
    2. Make sure you have pasted your **Gemini API Key** correctly into the GEMINI_API_KEY variable at the top.

3. **In Google AI Studio**, find your API key and click the link to its **"Google Cloud project"**.
4. In the Cloud Project, use the search bar to find **"Generative Language API"** and make sure it is **ENABLED**.
5. In the Cloud Project, go to the **"Billing"** section and make sure the project is **linked to an active billing account**.
6. After all this, you **must re-deploy** your Apps Script (see Step 3 of this guide).

- **Problem:** My generateAllPdfReports button gives a "No item with the given ID" error.
  - **Cause:** A Google Doc ID or Folder ID is wrong, or the script doesn't have permission.
  - **Solution:**
    1. **Re-Authorize:** Run the generateAllPdfReports function *manually* from the script editor once to grant it Drive/Docs permissions (see Step 3).
    2. **Check IDs:** Double-check the PDF_OUTPUT_FOLDER_ID and DEFAULT_REPORT_TEMPLATE_ID in your script (lines 13-16).
    3. **Check Checklists:** Make sure your Report Template ID column header is spelled *exactly* right and that the IDs in that column are for **Google Docs**, not Google