

Leave Request Automation System

Project Overview

A Leave Request Automation System built using Google Apps Script and an HTML-based web form to automate the employee leave application and approval process. The system integrates directly with Google Sheets and Gmail, eliminating manual tracking and email communication.

Key Features

Smart Web Form (Frontend)

- Developed using HTML, CSS, and JavaScript.
- Automatically fetches the logged-in user's email and pre-fills Employee ID, Name, and available Shift and Lead details.
- Supports multiple leave dates in a single submission.
- Prevents users from selecting duplicate or already-applied leave dates.
- Validates inputs dynamically and displays user-friendly alerts.

Backend Automation (Apps Script)

- Uses Google Apps Script to handle all server-side logic.
- Reads and writes data to multiple sheets — **Leads** and **Leave Requests**.
- Sends automated emails to employees and their leads upon submission and approval/rejection.
- Logs timestamps and statuses automatically.
- Prevents duplicate leave entries by cross-checking existing dates in the sheet.
- Dynamically builds CC email lists from the **Leads** sheet.

Automated Notifications (MailApp)

- Upon leave submission, sends confirmation emails to the employee and CCs to team leads.
- When the lead updates the **Status** column in Google Sheets (Approved/Rejected/Cancelled), the system:
 - Auto-triggers a timestamp.
 - Sends an automatic follow-up email with the updated status and leave details.

Scalable and Secure

- Deployed as a standalone web app.
- Access control managed through Google Workspace login.
- Stores all data within Google Sheets, enabling easy auditing and future dashboard integration.

Impact / Benefits

- Reduced manual work for HR and leads by 80% through automated leave requests and follow-ups.
- Eliminated human errors in tracking and duplicate submissions.
- Ensured real-time transparency of employee leave records.
- Centralized all data in a single Google Sheet with automated email notifications, removing the need for manual email tracking.

Mortgage Document Correction Automation Tool

Project Overview

A Mortgage Document Correction Automation Tool built using Google Apps Script, HTML, CSS, and the Gmail API to streamline and automate the correction email workflow for mortgage funding and post-closing teams. The web app eliminates repetitive manual work by dynamically generating pre-filled Gmail drafts for correction requests based on loan details, borrower information, and document types.

Key Features

Dynamic Web App Interface

- Developed a clean, responsive HTML form hosted as a Google Web App.
- Users select the correction type (Funding, Flip, Mavent, or ASAP) and enter loan details such as Loan Number, Borrower Last Name, and Property Address.
- The app automatically loads "To" and "CC" email options from a central **EmailConfig** Google Sheet.

Integrated Google Sheets Configuration

- Reads live configuration data (To, CC, intro, and closing templates) from a sheet named **EmailConfig**.
- Enables non-technical users to update email recipients and templates directly from Google Sheets without modifying code.

Automated Gmail Draft Creation

- On form submission, the app composes a professional email body containing borrower and document correction details.
- Automatically attaches pasted snip images and uploaded PDFs.
- Creates a Gmail draft (not sent directly) with all recipients, attachments, and formatted HTML content prefilled.

Error Prevention and Validation

- Implements front-end JavaScript validation to prevent incomplete or invalid submissions.
- Checks for empty fields, invalid email formats, and ensures only valid attachments are accepted.
- Displays real-time feedback messages such as "Draft Created" or "Error" for a professional user experience.

Supports Multiple Workflow Scenarios

- Handles multiple correction workflows: Funding, Flip, Mavent, and ASAP.
- Each workflow uses its own recipient group and email template.
- Automatically sets subject lines based on workflow type:
 - **Urgent Funding Delay:** BorrowerName, Address, Loan#
 - **Post Closing Issue:** BorrowerName, Address, Loan#
 - **Mavent Fail:** #LoanNumber
 - **Need to Fund ASAP:** #LoanNumber

Attachments & Inline Images Handling

- Allows users to paste snips directly as screenshots or document snippets.
- Converts pasted images to inline (cid-based) images embedded within the email.

- Supports multiple PDF uploads and automatic attachment to drafts.

Output

- After submission, displays a confirmation message and provides a direct Gmail draft link.
- Opens the draft instantly for review and sending, saving several minutes per correction.

Impact & Benefits

- Reduced manual correction email drafting time by **70–80%**.
- Ensured accuracy and consistency in recipient emails and templates.
- Enabled real-time updates via Google Sheets without technical intervention.
- Simplified attachment handling with drag-and-drop and paste functionality.
- Increased team productivity and improved communication transparency.

Mortgage Escalation & RCA Automation System

Overview

A fully automated Escalation and RCA (Root Cause Analysis) Management Tool developed using Google Apps Script, HTML, CSS, JavaScript, and the Gmail API, integrated with Google Sheets. The system automates the entire workflow of raising, tracking, and resolving mortgage escalations by sending structured escalation emails, tracking RCA submissions, and sending automated reminders — eliminating manual follow-up.

System Workflow

1. Escalation Form

- A user-friendly Google Web App form enables team members to submit loan escalations with key details such as Loan Number, Processed Date, Name, Escalation Type, and Error Description.
- Includes a **Category** dropdown (Production, Audit, Live Audit).
- Supports direct screenshot/snippet pasting (Ctrl+V) for inline image uploads into Gmail.

On submission:

- Details are logged in the **Entry Sheet** with a "Pending" status.
- A formatted escalation email is automatically sent to the assignee containing:
 - Loan details in a professional HTML table
 - Inline-embedded snips
 - A "Click Here" link for RCA submission
 - Auto-added CC list from the **Users Sheet**

2. RCA Form

- The receiver (Team Lead or Analyst) clicks the "Click Here" link to open the **RCA Submission Web App**.
- The form captures:
 - RCA explanation
 - Key observations/findings
 - Corrective action plan

On submission:

- RCA data is logged in the **RCA Sheet** with date and time.
- The original escalation's status automatically updates from "Pending" to "Provided."
- A reply-all email is sent within the original Gmail thread confirming RCA resolution.

3. Automated RCA Reminder System

- A time-based trigger (cron job) runs daily or weekly.
- The system checks the **Entry Sheet** for any "Pending" RCAs.
- For each pending loan:

- Automatically replies in the same Gmail thread, reminding the user to provide RCA.
- Maintains contextual communication within the same thread.
- Sends a personalized reminder email with a clickable RCA submission link.

Impact

- Reduced manual RCA follow-ups by approximately **80%**.
- Centralized tracking of all escalations and RCAs.
- Improved accountability and ensured no missed RCA submissions.
- Automated reminders eliminated manual chasing.
- Seamless integration with Gmail and Google Sheets.
- Provided real-time visibility for management through **Entry** and **RCA** sheets.

Mortgage Funding Delay Tracker

Project Overview

An automation script built using Google Apps Script that integrates Gmail and Google Sheets to automatically log all funding delay email conversations for each loan number. The system reads loan data from the **Trail** sheet, copies it to a **Trailing** log, and fetches all related email threads from Gmail — saving hours of manual searching and ensuring that all communication history is easily traceable for audit or follow-up.

How It Works

1. Data Logging from 'Trail' to 'Trailing'

- Whenever a new row is added in the **Trail** sheet, the script automatically copies columns **A:H** into the **Trailing** sheet.
- This creates a permanent log entry that serves as a tracking base for that loan number.

2. Automated Gmail Conversation Extraction

- A time-driven Apps Script function runs daily.
- It scans the loan numbers (column **D**) in the **Trailing** sheet.
- For each loan number, it performs a Gmail search using the query:
subject:"Urgent Funding Delay" subject:"<loanNo>" cc:me
- Retrieves all email threads related to that loan's funding delay communication.

3. Conversation Cleanup and Storage

- The script extracts each message's:
 - Sender name
 - Timestamp
 - Message content (plain text)
- Removes unnecessary quoted text, signatures, and formatting for readability.
- Appends the cleaned-up conversation summary into **Column I** of the same row in the **Trailing** sheet.

Impact

- Reduced manual searching for email threads by approximately **90%**.
- Ensured every funding delay conversation is centrally logged and auditable.
- Improved visibility for management on communication timelines.
- Created a single source of truth between Gmail and Google Sheets.
- Enhanced efficiency and transparency in the loan funding escalation process.