A Guide to Building a Complete AI Birthday Agent

This is the complete, start-to-finish guide for setting up your **Two-Stage AI-Powered Birthday Notification System** using Google Sheets, Google Apps Script, and the **Gemini API**.

We will do it in 2 build Modules. In Module 1 we will create an automation app that generates a **unique, context-aware message** for each person, and sends a single, consolidated reminder email directly to your inbox on the person's birthday. In Module 2 we will upgrade this to a Birthday Agent that also send gift suggestions 2 weeks before the birthday.

Module 1

The System Overview: Your Daily Sentiment Engine

This project, which we'll call the **AI-Powered Birthday Notification System**, relies on three core components:

1. **Google Sheets (The Data):** Where you store the person's name, birthday, and personalized notes (context).
2. **Google Apps Script (The Engine):** A JavaScript environment that runs on a schedule, reads your Sheet, and handles the emailing logic.
3. **Gemini API (The Intelligence):** Generates a custom, heartfelt message using the context provided in your Sheet.

Phase 1: Data Foundation (Google Sheets)

Your spreadsheet is the brain of the operation, holding all the data necessary to trigger personalized messages.

Step 1: Create and Name Your Sheet

1. Go to Google Sheets and create a new spreadsheet. Name it **Birthday AI Reminder**.
2. Rename the first tab (sheet) to exactly: **Birthdays**. *This name must match the code exactly!*

Step 2: Define and Input Your Data

Enter the following column headers in the first row (Row 1) and populate the data.

Column	Header	Example Value	Purpose
A	**Name**	Jane Doe	The person's full name.
B	**Birthday (MM/DD)**	11/25	The month and day (MUST include the leading zero).
C	**Classification**	3	A number for the relationship category.
D	**Last Sent Year**	2024	**The code manages this!** Keeps track of when the last email was sent.
E	**Notes/Context**	Loves gardening and has a pet cat named Miso.	Context used by the AI to create a unique message and gift suggestions
F	**Last Gift Sent Year**	2024	Tracks if the *gift reminder* was sent this year. (this is for Module 2)

Classification Key (Column C) for Message Tone:

- **1:** Close Family
- **2:** Extended Family
- **3:** Close Friends
- **4:** Friends
- **5:** Deceased (Triggers a reflective, gentle message)

Phase 2: Code Integration (Google Apps Script)

We will use Google Apps Script to write the automation engine.

Step 3: Open Apps Script

1. In your Birthday AI Reminder spreadsheet, go to **Extensions** > **Apps Script**.
2. This opens the code editor. Delete any placeholder text in the Code.gs file.

Step 4: Paste the Complete Code (in attached document)

Copy the entire working script from the Birthday Automation Code PDF and paste it into the empty Code.gs file.

Important: Before saving, find the line const RECIPIENT_EMAIL = "CHANGE THIS TO YOUR EMAIL ADDRESS"; and replace the placeholder with your actual email address.

Step 5: Set Your Project Time Zone

1. In the Google Apps Script editor, look for the **Project Settings** (gear icon ⚙️) on the left sidebar.
2. Under the **General settings** section, you will see a field labeled **Time zone**.
3. Click on the current time zone setting and select your desired time zone from the dropdown menu. **You must choose a specific time zone** (like *America/New_York* or *Europe/London*) for the daily trigger to fire at the correct local time (e.g., 7:00 AM to 8:00 AM in your chosen zone).
4. Once selected, the time zone is automatically saved.

Phase 3: Configuration (API Key & Authorization)

This phase connects your script to the Gemini AI model.

Step 6: Get Your Gemini API Key

1. Go to **Google AI Studio** (search for "Gemini API key").
2. Follow the prompts to select or **create a Google Cloud Project** to house your key.
3. Click the button to **"Create API key."**
4. **Copy the entire key string immediately.**

Step 7: Paste the API Key into Script Properties

1. In your Apps Script editor, go to the **Project Settings** (gear icon) on the left sidebar.
2. Scroll down to **Script properties** and click **Add script property**.
3. For **Property**, enter: GEMINI_API_KEY
4. For **Value**, paste the entire key you copied in Step 5.
5. Click **Save script properties**.

Step 8: Initial Authorization (Grant Permissions)

The script needs permission to read your sheet, connect to the internet (Gemini API), and send emails.

1. In the Apps Script editor, select the function **sendBirthdayReminders** from the dropdown menu at the top.
2. Click the **Run** button (triangle icon).
3. A pop-up will appear asking you to **Review permissions**. Click it.
4. Follow the prompts to select your Google account, click **Advanced**, then **Go to [Project Name] (unsafe)**, and finally **Allow** all permissions (email, sheet, and external connection).

Phase 4: Launch and Automation (The Daily Trigger)

This final step schedules your app to run every day.

Step 9: Create the Daily Trigger

1. In the Apps Script editor, select the function **createDailyTrigger** from the dropdown menu at the top.
2. Click the **Run** button.

Once executed, this function sets up the schedule. Your app will now automatically run every day between **7:00 AM and 8:00 AM** to check for birthdays and send you a consolidated email with the AI-generated messages!

Troubleshooting & Debugging

If your email arrives but the AI message says **"AI message generation failed,"** this is usually a hidden API configuration issue.

1. **Run the testApiKey function:** In the Apps Script editor, switch the function selector to testApiKey and click Run.
2. **Check Logs:** Go to **Execution Logs** and look for API TEST RESULT: SUCCESS!. If it says **FAILURE**, the issue is the key, billing, or quota.
3. **Final Fix:** If the test key works, the issue is often a slight API request structure problem. Since we fixed that, running the script again should resolve it. If the issue persists, carefully verify that your GEMINI_API_KEY in the Script Properties is active and correct.
4. Note you can open Gemini and use it to debug any issues

You now have a fantastic automated system: the AI-Powered Birthday Notification System. It runs daily, checks your spreadsheet, and generates a personalized birthday message using the Gemini API.

Module 2

We can transform this passive reporter into a simple AI Agent that solves the next most common problem: *What gift should I buy?*

By giving the system a second, complex task, generating thoughtful gift recommendations based on the person's interests we dramatically increase its value and push it into agent territory.

We're moving beyond simple automation. This guide helps you build a sophisticated **AI Agent**, a tool that actively manages your key relationships by performing two critical, contextual tasks on two different days: sending gift suggestions 14 days early, and providing a personalized message on the birthday.

This is the complete, start-to-finish guide for upgrading to your **Two-Stage AI-Powered Birthday Notification System** using Google Sheets, Google Apps Script, and the **Gemini API**.

Phase 1: Data Foundation

You built the spreadsheet in Module 1

Classification Key (Column C) and Agent Logic:

- 1 (Close Family) & 3 (Close Friends): Receives AI Gift Suggestion (14 Days Early) + AI Message (On Birthday).
- 2 (Extended Family) & 4 (Friends): Receives AI Message Only (On Birthday).
- 5 (Deceased): Receives AI Reflective Message Only (On Birthday).

Phase 2: Code Integration (Google Apps Script)

This is the logic that executes the two-stage check and the two parallel API calls.

Step 10: Open Apps Script

1. In your Birthday AI Reminder spreadsheet, go to **Extensions** > **Apps Script**.
2. This opens the code editor. Delete any placeholder text in the Code.gs file.

Step 11: Paste the Complete Agent Code

Copy the complete, working script from the Birthday Agent Code PDF and paste it into the empty Code.gs file.

Important: Before saving, find the line const RECIPIENT_EMAIL = "CHANGE THIS TO YOUR EMAIL ADDRESS"; and replace the placeholder with your actual email address.

$$Insert the complete ` birthday-notifier-two-stage-agent.js` file content here$$

Phase 3: Launch and Automation (The Daily Trigger)

This final step schedules your Agent to run every day.

Step 12: Create the Daily Trigger (CRUCIAL STEP)

Since we renamed the main scheduler function to mainDailyCheck() to handle the two-stage logic, you must run the trigger setup function again to update your schedule.

1. In the Apps Script editor, select the function **createDailyTrigger** from the dropdown menu at the top.
2. Click the **Run** button.

This will delete any old triggers and set the script to run the **mainDailyCheck** function every day between **7:00 AM and 8:00 AM**. Your Agent is now fully operational!

Troubleshooting & Debugging

- **"AI generation failed":** The API key or connection failed. Run the testApiKey function (select it in the dropdown and click Run) and check the logs. If it reports failure, re-verify your key.
- **"No Gift Suggestions":** Check the **Classification** (Column C) in your sheet. Only 1 (Close Family) and 3 (Close Friends) receive gift ideas.
- **"I got the gift email, but not the birthday message":** Check that you entered a date into the **Last Sent Year** column (D) for the **birthday message** to be sent, and a date into the **Last Gift Sent Year** (F) for the **gift suggestion** to be sent. The Agent needs to know when the last event occurred for both.
- **Note you can open Gemini and use it to debug any issues**

And there you have it - a complete Birthday Agent.