Great idea! To log detection events directly to a Google Sheet (instead of an offline Excel file), you’ll need to use the Google Sheets API. Here’s how you can do it:

**Plan**

1. **Set up a Google Cloud project and enable the Google Sheets API.**
2. **Create service account credentials and download the JSON key.**
3. **Share your target Google Sheet with the service account email.**
4. **Install the required Python library: gspread and google-auth.**
5. **Modify your script to log events to the Google Sheet instead of Excel.**

**Step-by-step Instructions**

**1. Google Cloud & API Setup**

* Go to the Google Cloud Console.
* Create a new project (or use an existing one).
* Enable the **Google Sheets API** for your project.
* Go to **APIs & Services > Credentials**.
* Click **Create Credentials > Service account**.
* After creating, go to the service account, click **Add Key > Create new key > JSON**.
* Download the JSON file and save it in your project directory (e.g., service\_account.json).

**2. Share Your Google Sheet**

* Create a new Google Sheet (or use an existing one).
* Copy its URL (you’ll need the sheet ID).
* Share the sheet with the service account email (found in your JSON file, looks like xxxx@xxxx.iam.gserviceaccount.com) with **Editor** access.

**3. Install Required Libraries**

Run this in your terminal:

sh

Apply to package inst...

pip install gspread google-auth

**4. Modify Your Script**

Below is a code snippet to replace the Excel logging with Google Sheets logging.**You only need to replace the Excel-related functions and their calls.**

**Google Sheets Logging Functions**

python

Apply to package inst...

*import* gspread

*from* google.oauth2.service\_account *import* Credentials

*from* datetime *import* datetime

*# Google Sheets setup*

def init\_gsheet(*sheet\_name*, *creds\_path*):

*# Define the scope*

    scope = [

        "https://www.googleapis.com/auth/spreadsheets",

        "https://www.googleapis.com/auth/drive"

    ]

    creds = Credentials.from\_service\_account\_file(creds\_path, *scopes*=scope)

    client = gspread.authorize(creds)

*# Open the sheet by name (or by URL or ID)*

    sheet = client.open(sheet\_name).sheet1

*# If empty, add headers*

*if* not sheet.get\_all\_values():

        sheet.append\_row(["Name", "Timestamp", "Access"])

*return* sheet

def append\_gsheet\_log(*sheet*, *name*, *access*):

    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    sheet.append\_row([name, timestamp, access])

**How to Use in Your Main Script**

* At the start of your main() function, initialize the sheet:

python

Apply to package inst...

    creds\_path = os.path.join(script\_dir, "service\_account.json")

    sheet\_name = "Your Google Sheet Name"  *# Or use the sheet ID*

    sheet = init\_gsheet(sheet\_name, creds\_path)

* Replace all append\_excel\_log(log\_path, name, access) calls with:

python

Apply to package inst...

    append\_gsheet\_log(sheet, name, access)

**Summary**

* You’ll now have a live-updating Google Sheet with the same columns: Name, Timestamp, Access.
* You can view and share this sheet online, and it will update in real time as your program runs.

**If you want, I can generate the exact code changes for your script. Just let me know your Google Sheet name or ID, and confirm you have the service account JSON file ready!**

apply all these change to this program

Restore checkpoint

face\_recognition\_excel\_out.py

+39-23

Your program is now set up to log detection events directly to a Google Sheet instead of an offline Excel file.**What you need to do:**

1. **Install the required libraries:**

sh

Apply to package inst...

   pip install gspread google-auth

1. **Download your service\_account.json** from Google Cloud and place it in the same directory as your script.
2. **Set your Google Sheet name or ID:**

* Replace "YOUR\_GOOGLE\_SHEET\_NAME\_OR\_ID" in the code with your actual Google Sheet name or the Sheet ID.
* Make sure your Google Sheet is shared with the service account email (found in your JSON file).

**How it works:**

* The program will append each detection event (Name, Timestamp, Access) to the first worksheet of your Google Sheet, live and online.

Let me know if you need help with the Google Cloud setup or want a ready-to-paste configuration block for your credentials and sheet!