# PC Activity Logger with Google Sheets

## 1. Project Overview

This project tracks activity on your PC by monitoring application processes and window titles, logging them live to a Google Sheet which can be accessed from your phone.

Features include:

- Real-time logging of newly opened apps
- Capturing active window titles and timestamps
- Remote STOP command from Google Sheet cell
- Data stored in a centralized Google Sheet

## 2. Tools & Libraries Used

Python Libraries:

- psutil: to monitor system processes
- pywin32: to get the foreground window title (Windows only)
- gspread: to interact with Google Sheets
- oauth2client: for Google API authentication

Online Accounts/Tools:

- Google Cloud Console
- Google Sheets

## 3. Setup Steps

Step 1: Install Required Python Libraries

Command:

```
pip install psutil pywin32 gspread oauth2client
```

Step 2: Create Google Cloud Project

- Visit: https://console.cloud.google.com/
- Create a new project

Step 3: Enable APIs

- Google Sheets API: https://console.cloud.google.com/apis/library/sheets.googleapis.com
- Google Drive API: https://console.cloud.google.com/apis/library/drive.googleapis.com

Step 4: Create Service Account

- Go to: https://console.cloud.google.com/iam-admin/serviceaccounts
- Create a new service account
- Assign "Editor" role
- Generate a JSON key and download it

Step 5: Share Your Google Sheet

- Create a sheet named: PC Activity Logger
- Share it with the service account email (from the JSON file)
- Give Editor permission

## 4. Full Script Code (With Remote STOP Feature)

```
import psutil
import win32gui
import time
from datetime import datetime
import gspread
from oauth2client.service_account import ServiceAccountCredentials

scope = ["https://spreadsheets.google.com/feeds", "https://www.googleapis.com/auth/drive"]
creds = ServiceAccountCredentials.from_json_keyfile_name("credentials.json", scope)
client = gspread.authorize(creds)
sheet = client.open("PC Activity Logger").sheet1

sheet.clear()
sheet.append_row(["Time", "Process Name", "Window Title"])
sheet.update('D1', [['RUNNING']])  # use [['...']] format

known_pids = set()

def get_active_window_title():
    try:
        return win32gui.GetWindowText(win32gui.GetForegroundWindow())
    except:
        return "N/A"

print("Monitoring started. Type 'STOP' in D1 to exit.\n")

while True:
    try:
        control = sheet.acell('D1').value.strip().upper()
        if control == 'STOP':
            print("STOP received from sheet. Exiting...")
            break
    except:
        pass

    for proc in psutil.process_iter(['pid', 'name']):
        pid = proc.info['pid']
        pname = proc.info['name']
        if pid not in known_pids:
            known_pids.add(pid)
            now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            title = get_active_window_title()
            try:
                sheet.append_row([now, pname, title])
            except Exception as e:
                print("Error logging to sheet:", e)
    time.sleep(3)
```

## 5. Common Errors & Fixes

Error: Google Drive API not enabled (403)

Fix: Enable it at https://console.developers.google.com/apis/api/drive.googleapis.com

Error: SpreadsheetNotFound (200)

Fix: Make sure your Google Sheet name is exact and shared with the service account

Error: APIError: Invalid value at 'data.values'

Fix: Use this format when writing to cells:

```
    sheet.update('D1', [['RUNNING']])   # Not just 'RUNNING'
```

## 6. Next Steps (Optional)

- Run this script silently at startup using Task Scheduler or a background script
- Add features: Pause/Resume, Save to local CSV, Auto-email logs
- Convert script into a .exe with auto-start capability