# Attendance Tracker

## Introduction

Attendance tracker is a simple Python program to track the attendance of students, mentors, coaches and volunteers to/from events for the purpose of being able to note hours given to our local community.

This tracker does NOT track any precise location data other than a generic tag such as “Workshop”, “RoughRiders 2025”, “Library June 2025” etc..

Data collected is stored locally on each ‘device’ that has been given access to the ‘Track Database’. Optionally the data can be stored on a Google Sheet workbook (to allow multiple ‘devices’ to access at the same time) and optionally in a github repository. Both Google Sheets access and github access should be configured accordingly to ensure the data is not exposed outside the organizational requirements. (i.e. DO NOT MAKE THE github repository public!!!). Encryption of the local and github data is currently on the list of things to do. The Google Sheet is purposely NOT encrypted in order to allow data processing within the Google environment but should have sufficient protections implicit in the google workflow.

The ‘Track Database’ contains a unique ID for each ‘trackable member’ along with their name (for display purposes) and optionally a photo.jpg display image.

### Features

The attendance tracker is meant to be a simple to setup/configure, simple to operate real time attendance tracking system for small organizations. The initial design requirements are as follows…

#### Primary features

1. Support 300+ members
2. Support multiple simultaneous ‘devices’ to allow multiple entry points to the facility
3. Be as quick and easy as possible to sign in/out
4. Somewhat ‘fun’ to interact with
5. Support RFID tags for quick processing
6. Support easy data processing for reports etc…
7. Provide basic administration features
8. Able to operate without additional accounts/servers/fees etc… (depending on features needed)
9. Be reasonable cheap to build (will work on Windows but also Raspberry Pi with LCD)
10. Be open source, expandable and available for anyone to implement

#### Secondary features

1. Support off site events (remote, portable device that can sync in real time or at a later time)

### Is not…

The attendance tracker is not meant to be a fully featured, all singing, all dancing access tracker. Other options available seem to cost many $100s (even $1000s!!) and/or require provider accounts, server costs etc… It is not massively complex, does not inherently provide any data analysis

## Software setup

If you are planning on using a stand alone Raspberry Pi with touch screen LCD first follow the “Build an Attendance Tracker” section before continuing with the following software installation and configuration.

### Python

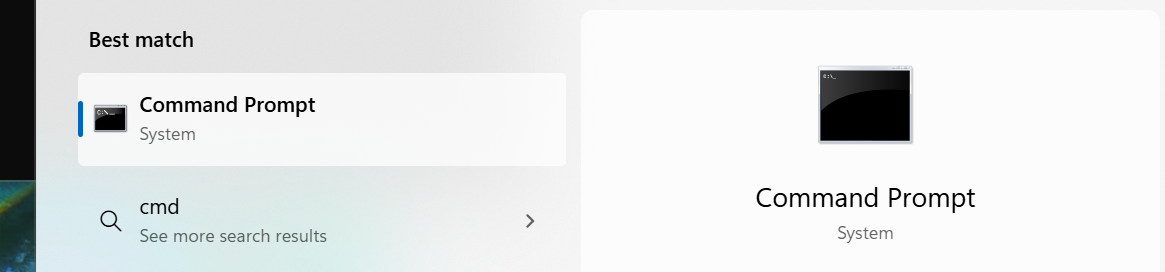
“Attendance Tracker” is written in the widely available and cross platform programming language “Python”. Most Linux systems, including the Raspberry Pi platform, will have Python installed by default, but Windows systems may not.

Whilst Attendance Tracker was developed primarily on a Windows computer, it was actually designed to run on a Raspberry Pi Zero W with an LCD touch screen and RFID card reader attached. This means that Attendance Tracker can actually be used on a Windows computer, but may not feel quite as integrated as when running on a Raspberry Pi system. Work is also in progress to make Attendance tracker run on both Android tablets and iPads.

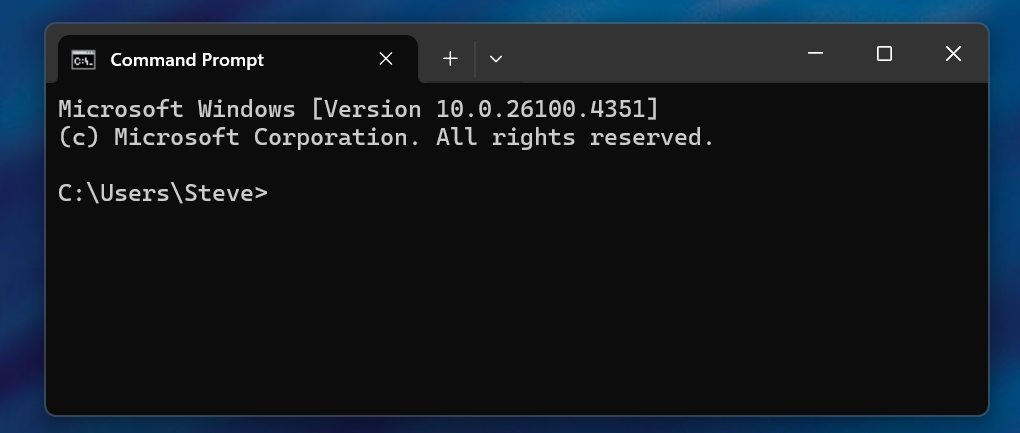
#### Installing Python on Windows (optional)

If you wish to run Attendance Tracker on Windows you can check if Python is installed by doing the following…

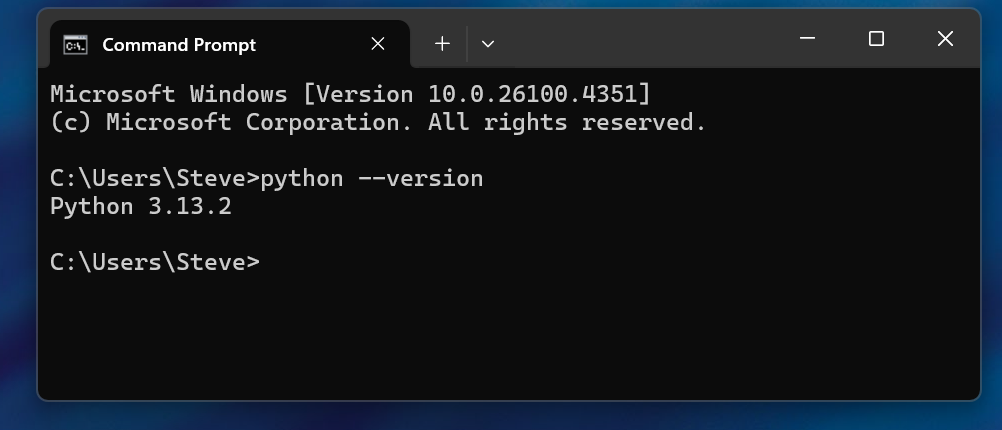
In Windows search type “cmd”. You should see the following appear…



Simply hit the enter key and the following should appear…



With this new window highlighted type “python –version”. If Python is installed then you should see something similar to the following…



This tells us that Python version 3.13.2 is installed.

Attendance Tracker requires Python version 3.10.7 or later.

If Python is not installed, or the version is lower than V3.10.7 then visit the Python community web page at <https://www.python.org>, select “downloads”, then download the latest version. Once downloaded run the installer and accept all the defaults unless you are familiar with Python.

#### Packages

Once Python has been installed it is necessary to add some additional software ‘packages’ that Python needs to be able to run Attendance Tracker. These packages are needed for things like graphics, serial port access, Google access etc…

On Raspberry Pi systems open a terminal window (ToDo)

On Windows highlight the command prompt window we recently opened. If you closed it simply re-open with the search we did earlier.

In the terminal/command window type the following…

“pip install pygame”  
“pip install pyserial”

### Google

#### Account

Create a Google account that will hold the database. If you do not want to store your database on Google Sheets then this step is not necessary. Note that without a Google Sheet managed database you will not be able to take advantage of the real time tracking/updates across multiple devices.

If you already have a Google account that you want to use for your tracking database then you can use that account. Be aware that there may be security concerns by linking the tracker application to your Google account that I am not aware of. I don’t think it is an issue, but I am not an expert.

For the Attendance Tracker to be able to talk to Google then the Google Sheet needs to be associated with a ‘Google Project’.

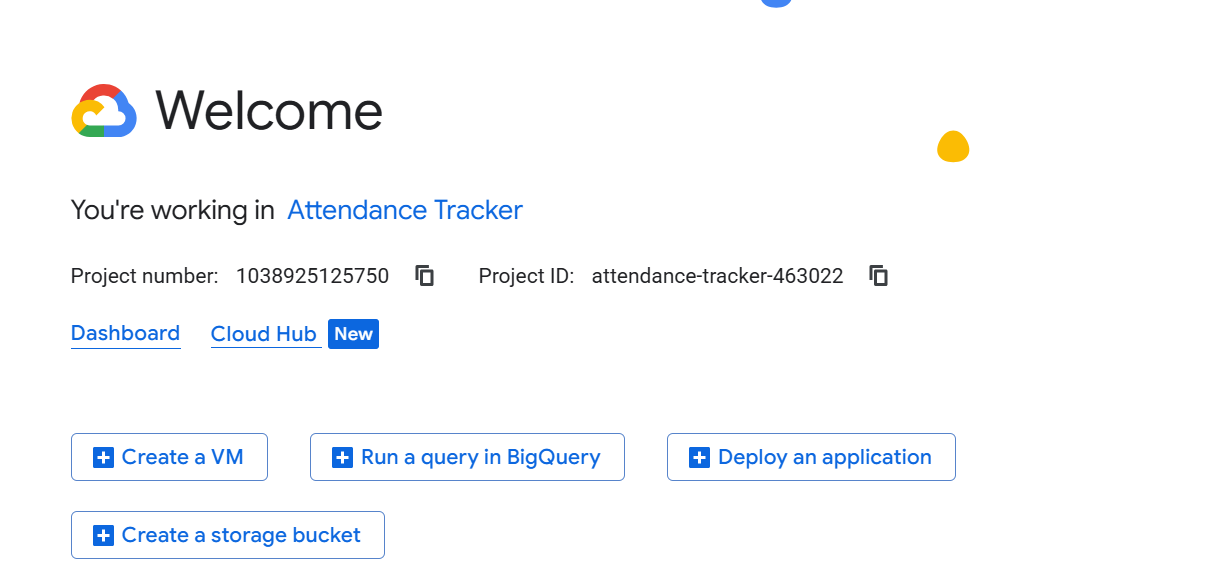
To create a project go to <https://console.cloud.google.com/projectselector2/>. I have not worked out how to get here by navigation from the mail account !!!

Click “Create Project” (Useful video for those interested…) <https://www.youtube.com/watch?v=zCEJurLGFRk>)

On the next window, for the Project Name use “Attendance Tracker”.

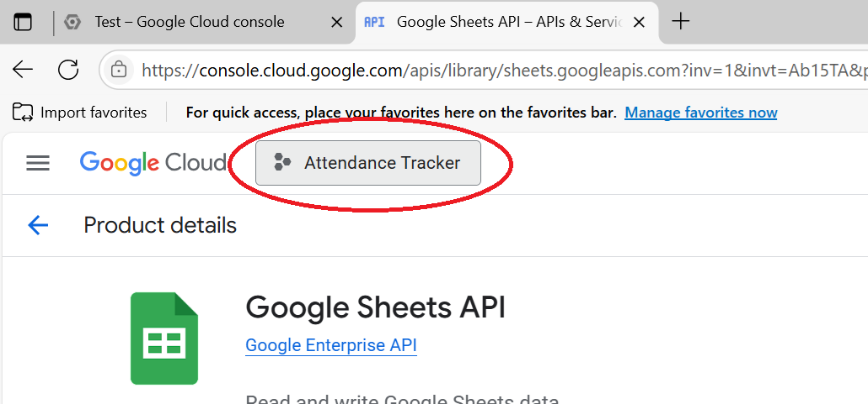
Click “Create”.

Next you should see a “welcome’ screen and indicate “You’re working in Attendance Tracker”.

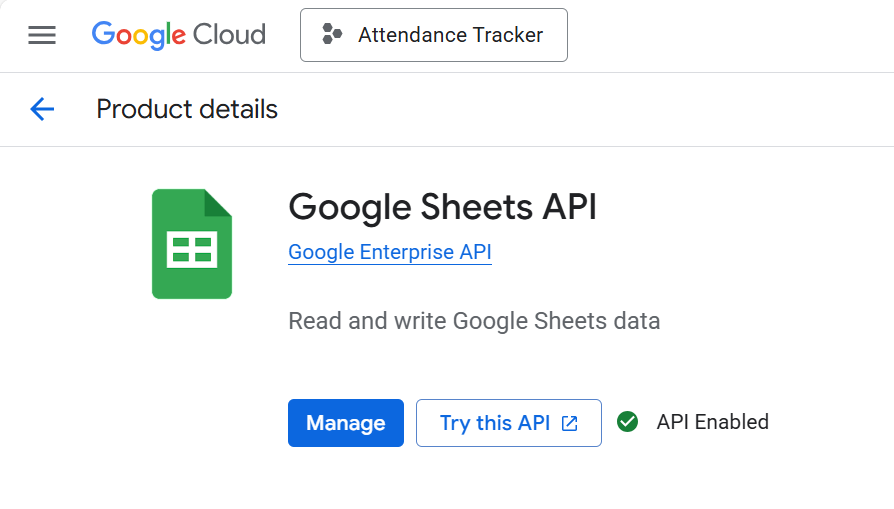


If not, try clicking on whatever is highlighted in blue after “You’re working in “ and a list of projects should appear. Click on “Attendance Tracker” to make it the active project.

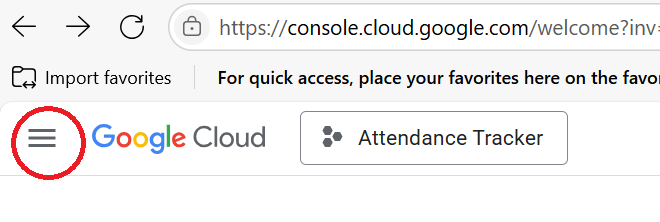
If all that fails then go to <https://console.cloud.google.com/apis/library/sheets.googleapis.com>, click on the active project selector in the top left and select “Attendance Tracker”.



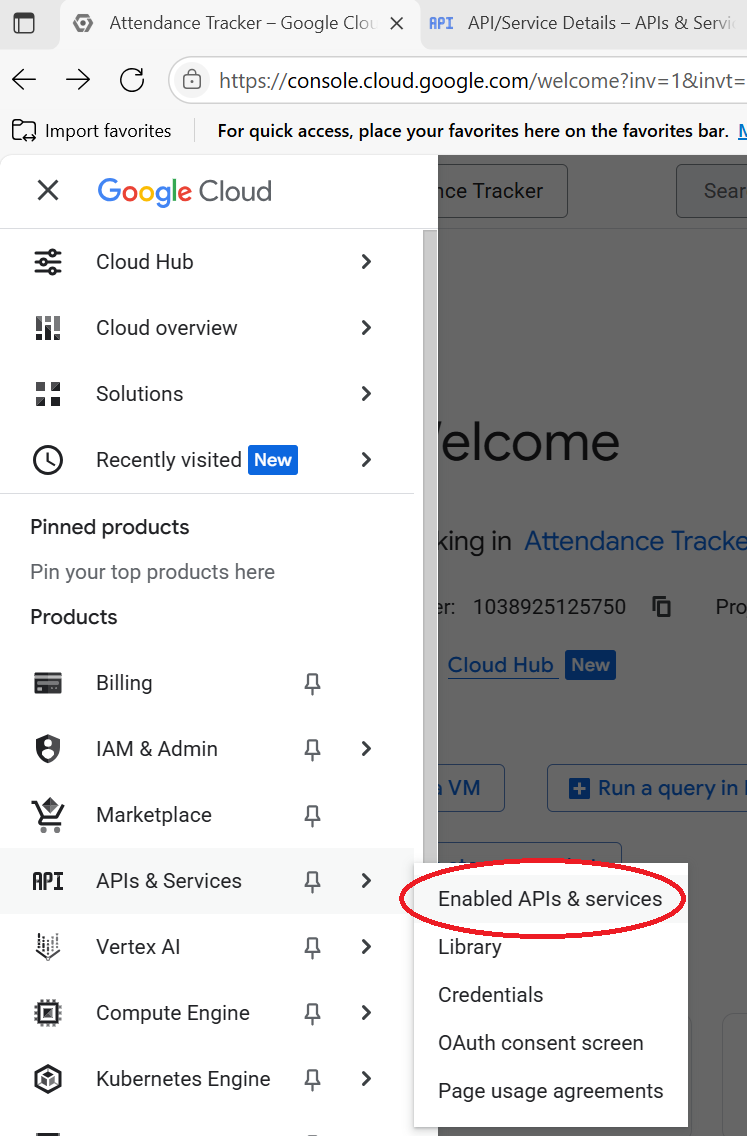
Then click “Enable”. Again, I have no clue how to get here through navigation !!! The next time you go to this link “Enable should have changed to “Manage”, indicating that the Sheets API access has been activated. Remember this link. It is VERY useful since navigation is so fundamentally broken in my mind!!



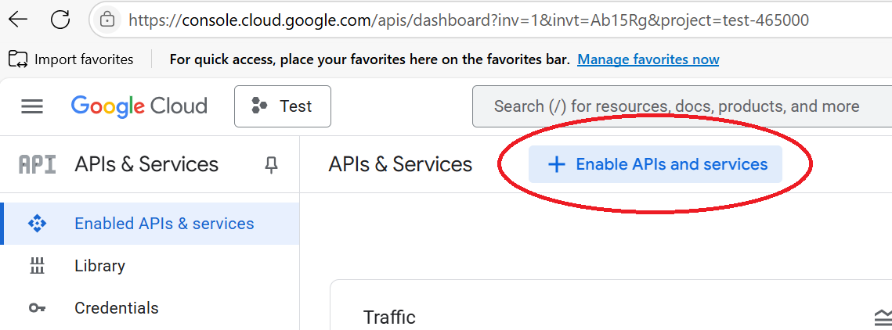
Next click the navigator icon and select “APIs & Services -> Enabled APIs & services”.



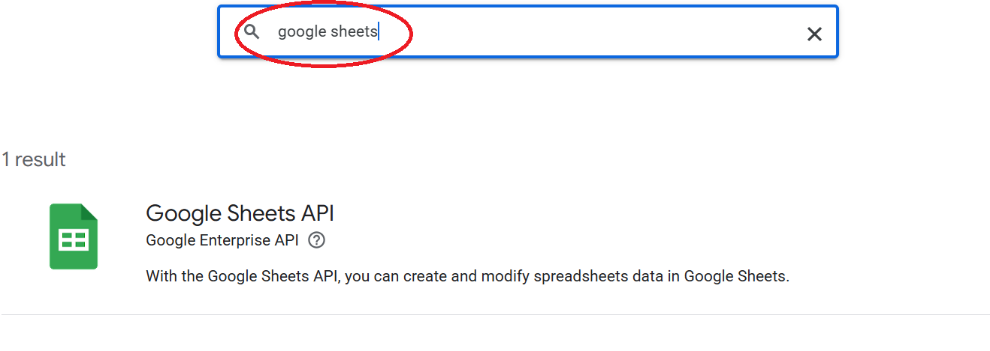
Then



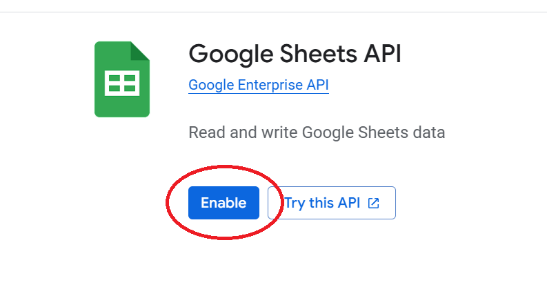
Next, click “Enabled APIs and services”.



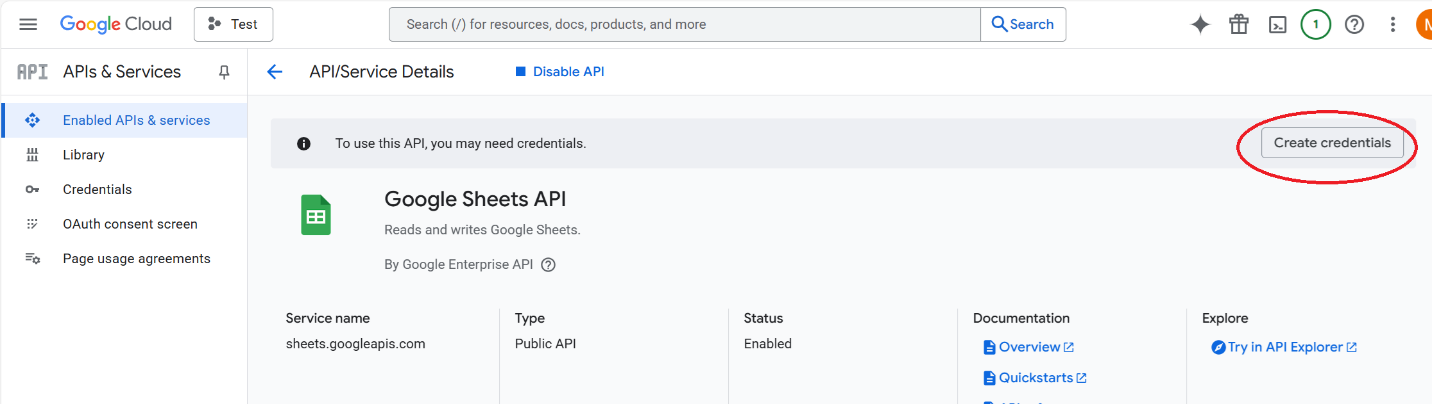
Type “Google sheets” in the search box and hit ‘Enter’, then click on the result.



Then click ‘Enable’.



You should then see a page similar to below.

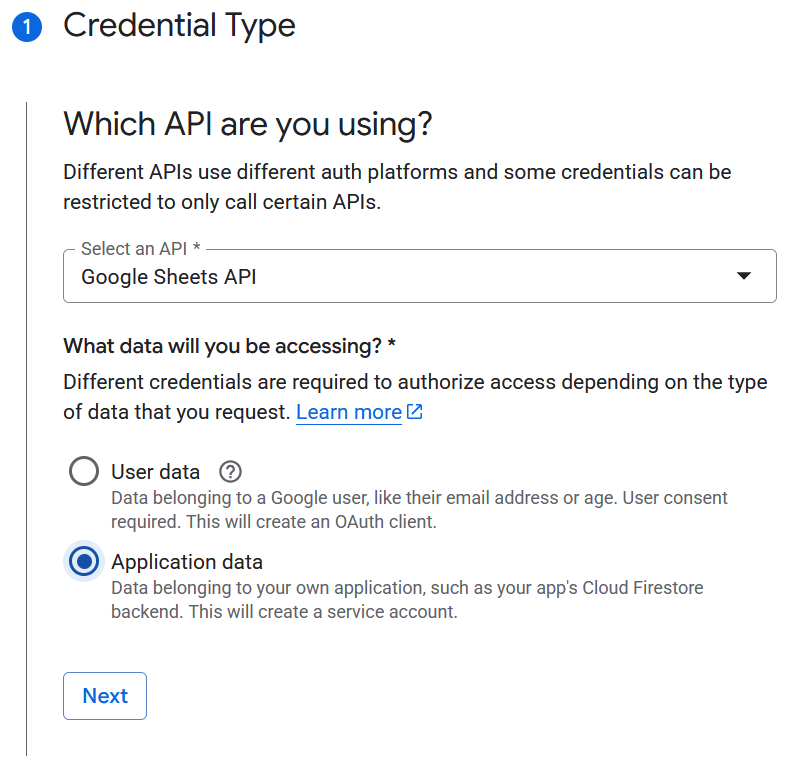


If you need to get back to this page at any time you should be able to navigate through the ‘Enable APIs & services’ flow above or use some of the links I provided above, remembering that “Enable” will have changed to “Manage”.

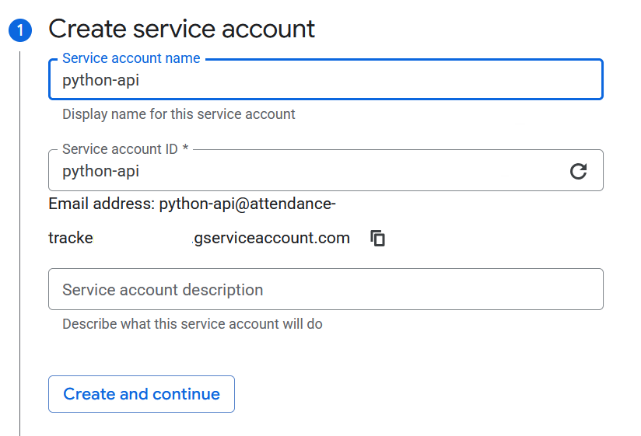
Next, we need to “Create Credentials”

On the page shown above, then click on “Create credentials” near the top right of the window.

Set the following, then click “Next”

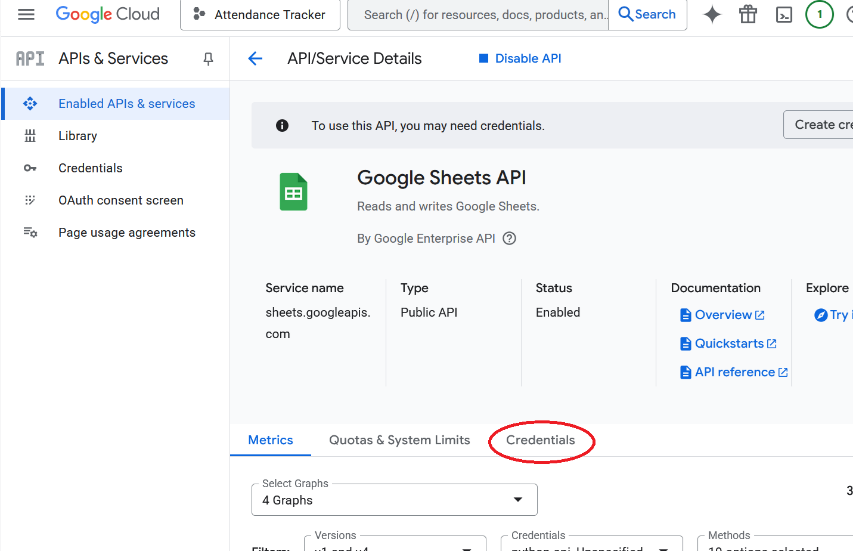


On the next screen set the following, then click “create and continue”

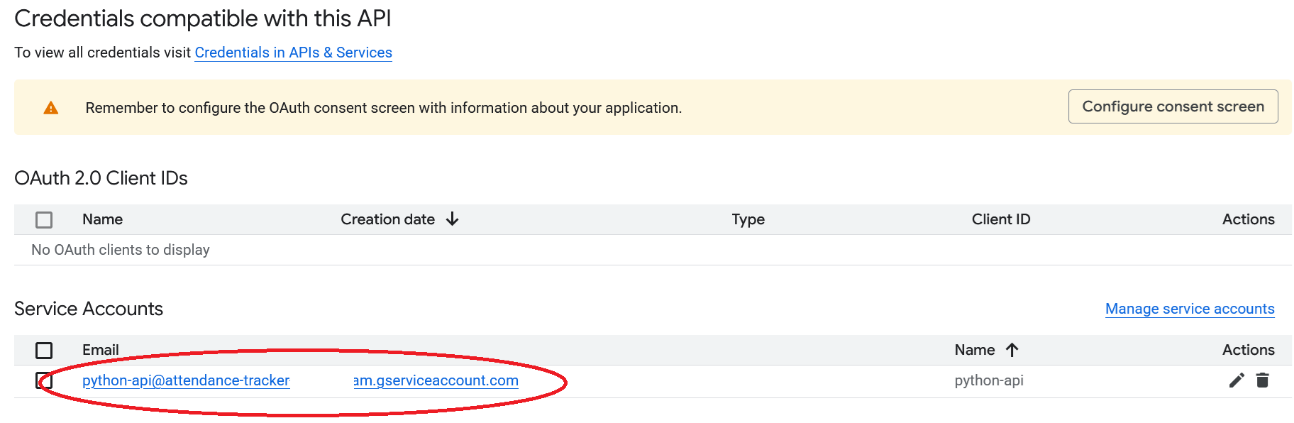


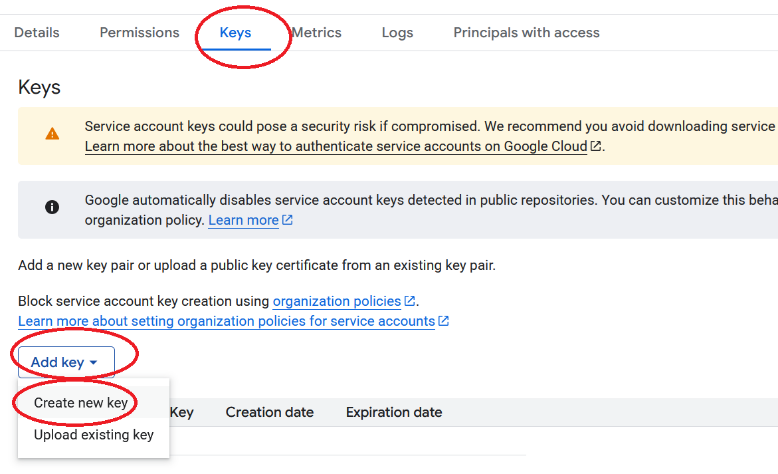
On the next screen in the “Select role” dropdown type “editor” then select the Editor entry from the drop down list” then click “continue” then finally “Done”.

Now we need to download the credentials we just created so click the “Credentials” tab in the middle of the window (NOT the Create Credentials we clicked earlier).

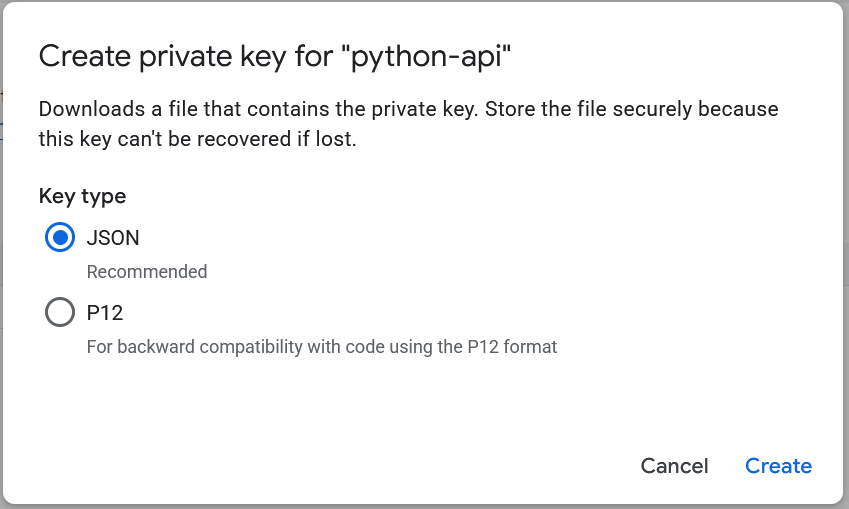


Then click on the “Service Account” we just created.



On the next screen click the tab “Keys”, then “Add key” and select “Create new key”

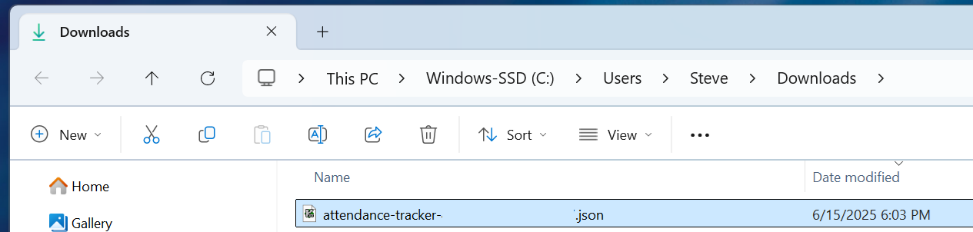
Ensure “JSON” is selected then click “Create”.



This will then save a file automatically to your computer. We need to copy this file to the same location where the Attendance Tracker Python file is located.

On Windows locate the file just downloaded (most likely it will be in your ‘downloads’ directory) and move it to the Attendance Tracker Python program location. Finally rename the file as “Credentials.json”.

NOTE : KEEP THIS FILE SAFE. DO NOT DISTRIBUTE IT TO ANYONE. THIS FILE CONTAINS THE KEYS TO ACCESS THE GOOGLE PROJECT WE JUST CREATED.



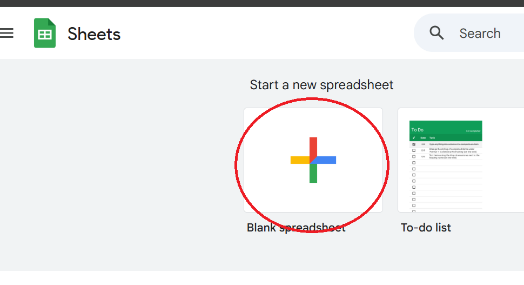
Now we need to link the Google Sheet Workbook with the Google Attendance Tracker service account.

#### Workbook

So, now we need to create a workbook called “AttendanceTracker”.

In a new web browser window/tab go to <https://workspace.google.com/products/sheets/> and sign in to the Google account you want to store the Attendance Tracker sheet in.

Start a new blank spreadsheet.

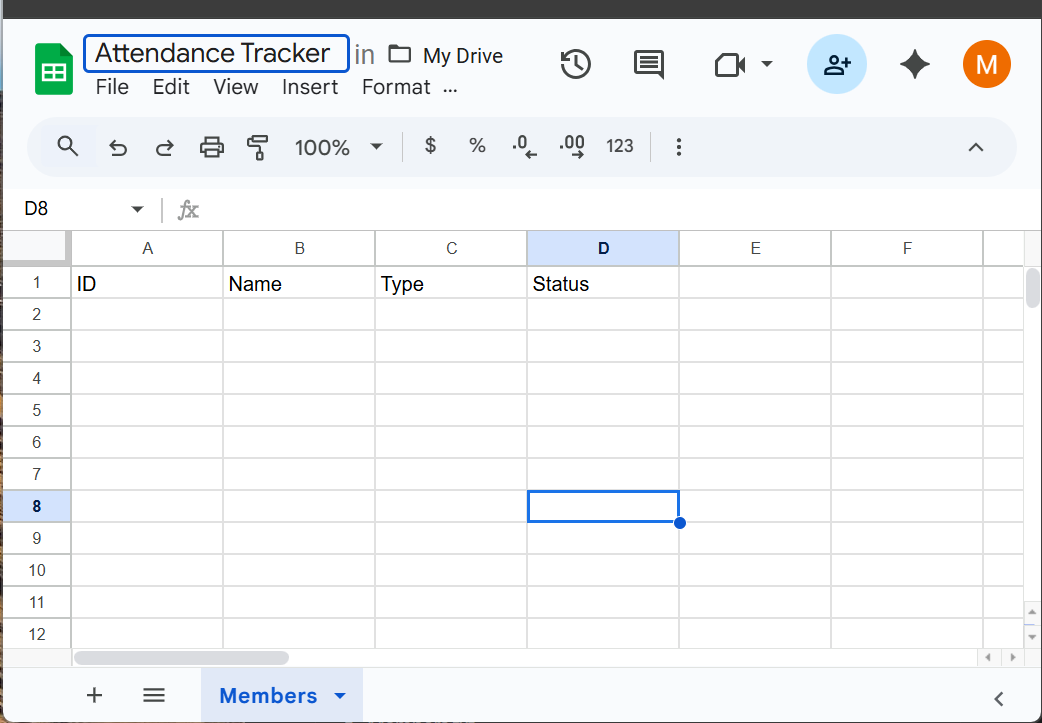


In this workbook name the first sheet “Members”.

On the “Members” sheet add the following column headers, starting at cell A1…

“ID”, “Name”, “Type”, “Status”

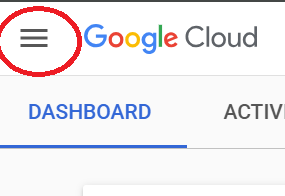
Double click on “Untitled spreadsheet” in the top left and rename the spreadsheet as Attendance Tracker.

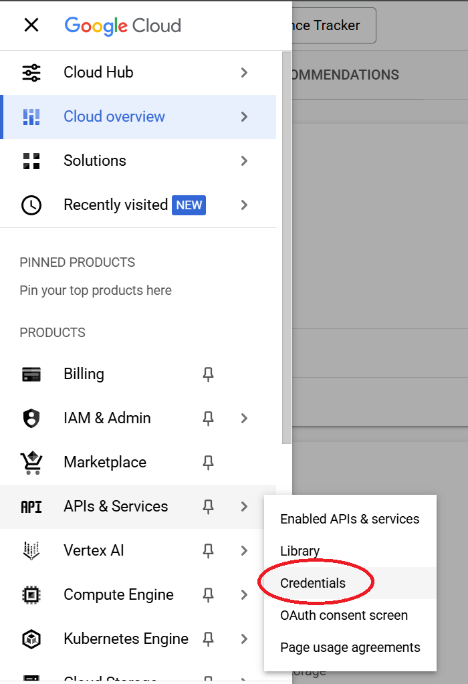


Eventually there will be a template workbook available to use as a starting point.

Now we need to ‘share’ this workbook with the service account so it can access it.

Back in the console.cloud window/tab click the top left navigation bars, then select APIs & services -> Credentials.

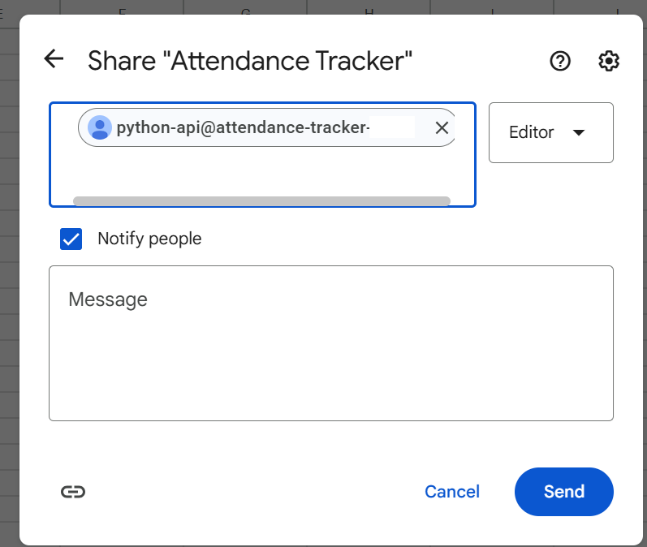




Under ‘Service Accounts” there will be at least one entry for ‘email’. If there are more than one then you are a power user and way ahead of me. Otherwise there should only be one email listed.

Select the email and copy it.

Go back to the Google Sheet window/tab with the sheet we just created and click ‘Share’ in the top right, then paste the email into the share text box. Make sure “Editor” is selected in the drop down to the right.

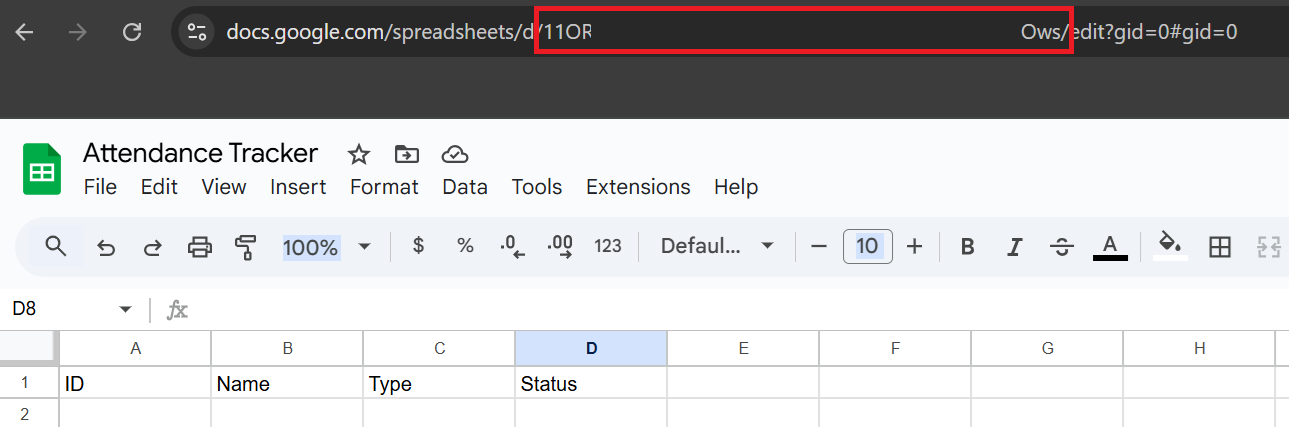


Congratulations. Attendance Tracker should now be able to talk to your Google Sheet.

Ahh, but…

There is one last piece of information we need from the Google sheet web page.

On the workbook window/tab you will see the web address for the sheet. We need to capture everything between “https://docs.google.com/spreadsheets/d/” and “/edit?gid=0#gid=0”



So in this example that would be “11O…………………Ows”.

Select this text, copy it and paste it into a text file called “config.txt” in the following format.

[SheetLocation] = “11O…………………Ows” Hardware

Different hardware options are planned for the attendance tracker, including an iPad app and an Android tablet app. Initially the hardware is based on a Raspberry Pi with a 7” touch screen.

## Build an Attendance Tracker

### Introduction

If you want a fairly cheap, stand alone, dedicated Attendance Tracker station then you can build one using a Raspberry Pi (Pi5 recommended although at a pinch even a Pi Zero2 W could be used, it just takes ages to boot and animations probably turned off!!).

The stations we built are composed of the following components and totaled approximately $150 each.

Raspberry Pi5 2GB $50.00

<https://www.microcenter.com/product/683269/raspberry-pi-5>

Pi5 fan $10.00

<https://www.amazon.com/dp/B0CZLPX2HC?psc=1&smid=A3B0XDFTVR980O>

Micro SD card (10 pack) $40.00

<https://www.amazon.com/gp/product/B07RVFZ3F3/ref=ox_sc_act_title_1>

12” HDMI to micro HDMI cable (2 pack) $13.00

<https://www.amazon.com/dp/B0CMTR9S3Y?ref=ppx_yo2ov_dt_b_fed_asin_title&th=1>

LIR2032 battery holder (6 pack) $9.00

<https://www.amazon.com/dp/B0D93Z7KD8?ref=ppx_yo2ov_dt_b_fed_asin_title>

LIR2032 rechargeable batteries (5 pack) $8.00

<https://www.amazon.com/dp/B0BBY5G57W?ref=ppx_yo2ov_dt_b_fed_asin_title>

USB Type-C 12” extension cable (2 pack) $10.00

<https://www.amazon.com/dp/B0D73CSC5C?ref=ppx_yo2ov_dt_b_fed_asin_title&th=1>

7” LCD touch screen $40.00

<https://www.amazon.com/dp/B0CRRB1GFN?ref=ppx_yo2ov_dt_b_fed_asin_title>

13MHz RFID reader (5 pack) $10.00

<https://www.amazon.com/dp/B0CCF4SNMF?ref=ppx_yo2ov_dt_b_fed_asin_title>

Raspberry Pi Pico Zero (3 pack) $12.00

<https://www.amazon.com/dp/B0CDWWH8HK?ref=ppx_yo2ov_dt_b_fed_asin_title>

Micro USB cable

Various 13MHz RFID tags

USB Type-C power supply (25W+)

Micro SD card reader/writer

Sticky foam pads $6.00

<https://www.amazon.com/dp/B0BCJWMGR1?ref=ppx_yo2ov_dt_b_fed_asin_title&th=1>

3D printed case

### Pi software setup

First we need to create an operating system SD card for the Raspberry Pi.

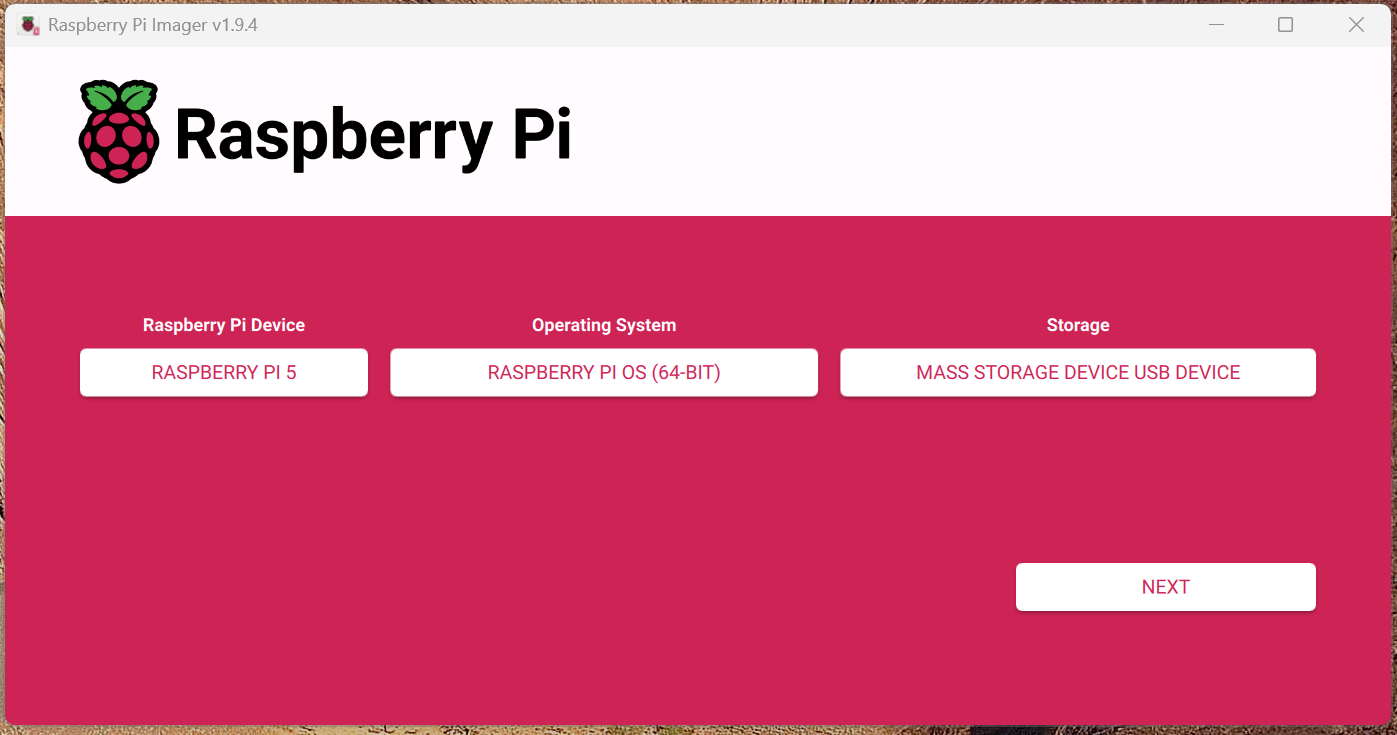
Side note: If you are planning on setting up multiple stations then the easiest way to achieve this is to start off configuring a station using the SMALEST SD card (e.g. 2G). Due to an awkward quirk of the Raspberry Pi ecosystem the SD card image we are about to create will automatically expand to fill the entire SD card when it is first used. Due to slight variations in SD card sizes (they are not all exactly, say, 4GB!!) it means you may not easily be able to copy one 4G card to another 4G card. You can, however always copy from a smaller card to a larger card, so make your ‘golden’ card as small as possible so you can easily copy to other SD cards later without any issues. Alternatively you can just go through this entire process for each station, but I am lazy!!

Go to <https://www.raspberrypi.com/software/> and download the Raspberry Pi imager.

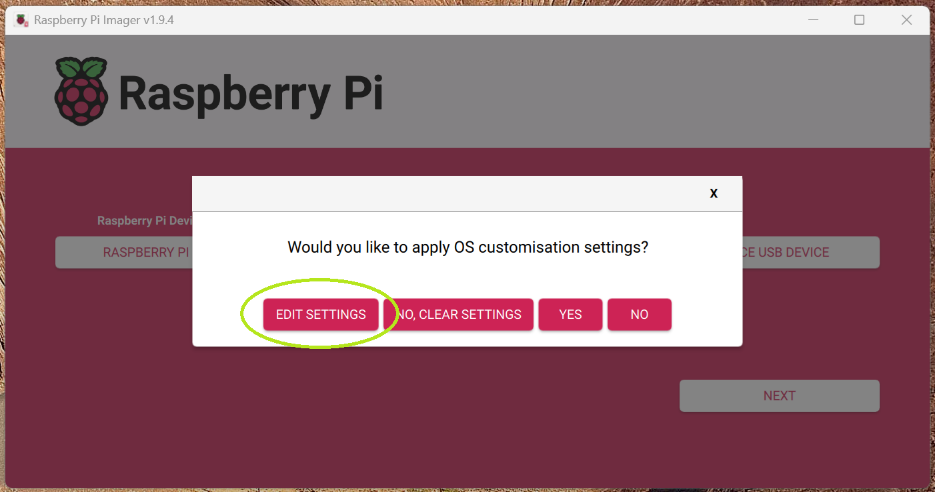
Run the image creator installer, then run the actual creator. Put an SD card in your SD card writer and plug in to your computer.

Set the following settings, then click “Choose Storage” to select your micro SD card.

MAKE SURE YOU SELECT YOUR MICRO SD CARD AND NOT ANY OTHER DRIVES YOU MIGHT HAVE CONNECTED TO YOUR COMPUTER!!! THIS IS VERY< VERY IMPORTANT!!! WHICHEVER DRIVE YOU SELECT WILL BE COMPLETELY REFORMATTED AND ALL EXISTING DATA DELETED IN A VERY PERMANENT WAY!!!



Click “Next” then select “Edit Settings”



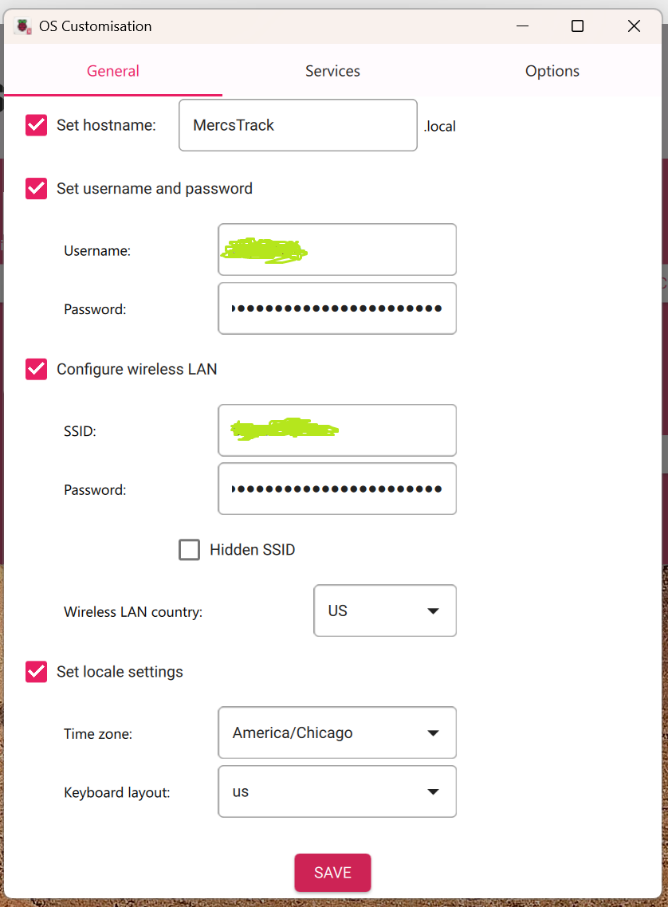
In the OS customization oage select the “General” tab and enter all the settings shown below.

The “hostname” is what this Attendance Tracker ‘station’ will be called on your network. If you have multiple stations then they should all have their own name. If you use the ‘copy SD card’ method for multiple stations then don’t forget to change their host names!

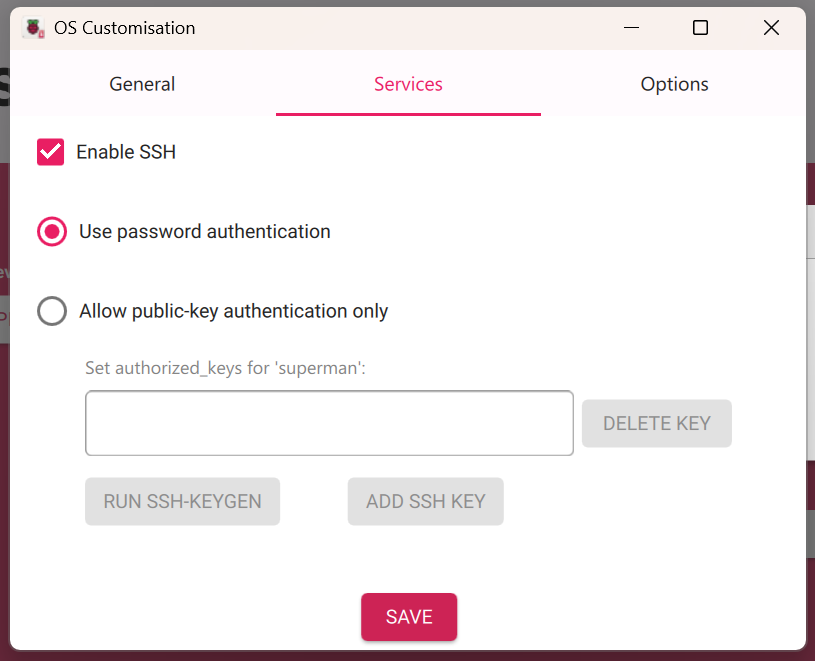
The username and password will be the credentials needed to log in to the Attendance Tracker station and will be needed to make changes and/or updates to the station. It is recommended that these credentials be the same for all Attendance Tracker stations, but as with all usernames and passwords they should be guarded with the utmost care and never divulged to anyone you don’t want to have access to not only the Attendance Tracker station, but your connected network.

To simplify the setup process it is also recommended to configure the wireless LAN, so enter the WiFi credentials for the network that this station will generally connect to. More can be added later if necessary (different locations, hotspots for remote access etc…).

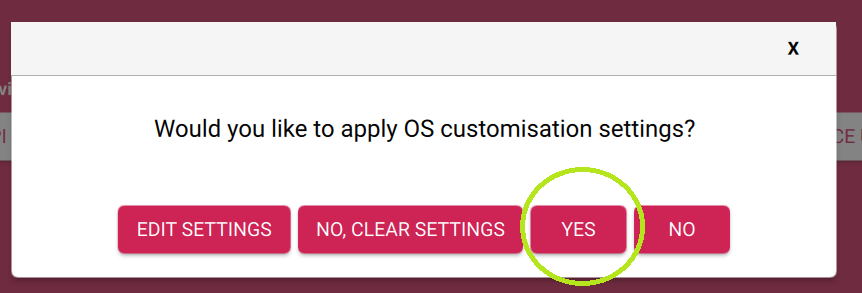
Set the locale settings to match your country, timezone and keyboard.



Next, click the “Service” tab and check “Enable SSH” with password authentication selected. This will make later configuration MUCH simpler.



Click “Save” then click “Yes”.



At the next dialog check that you really have the correct SD card inserted since it WILL be erased. Once 100%sure, click “Yes” again and wait for the SD card write and verify to complete.

Once the SD card write has been verified and is all good remove the SD card from your computer and plug in to your Raspberry Pi.

The first time you power up the Raspberry Pi it will reboot a few times. Do not worry, this is normal and should only happen the first time you power things up.

Once the Raspberry Pi is up and running you should then see a somewhat normal looking desktop on the LCD. Hopefully the WiFi connection worked too. If connected then you should momentarily see a message in the top right corner of the desktop indicating the “IP Address” that the Raspberry Pi was given. Make a not of this number. It is usually in the form ‘192.168.0.xxx’ where xxx can change occasionally.

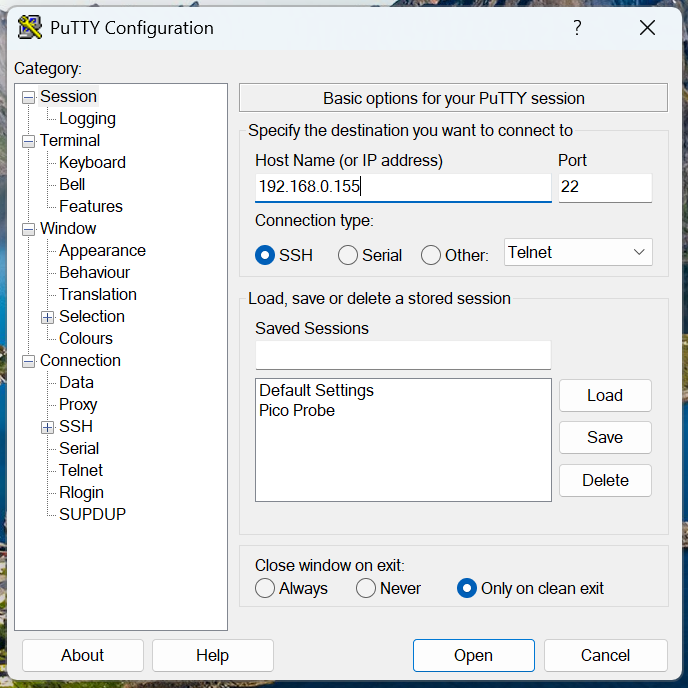
If the WiFi is not connected then you can connect a mouse and keyboard to the Raspberry Pi, click on the WiFi icon in the top right of the desktop and connect to your desired WiFi.

Once you are connected to your WiFi we can use your ‘regular’ computer on the same WiFi network to complete the remaining setup.

On your ‘normal’ computer we now need to install a ‘terminal’ program. On Windows the recommended program is called “Putty”. Putty can be downloaded from here <https://www.putty.org/>

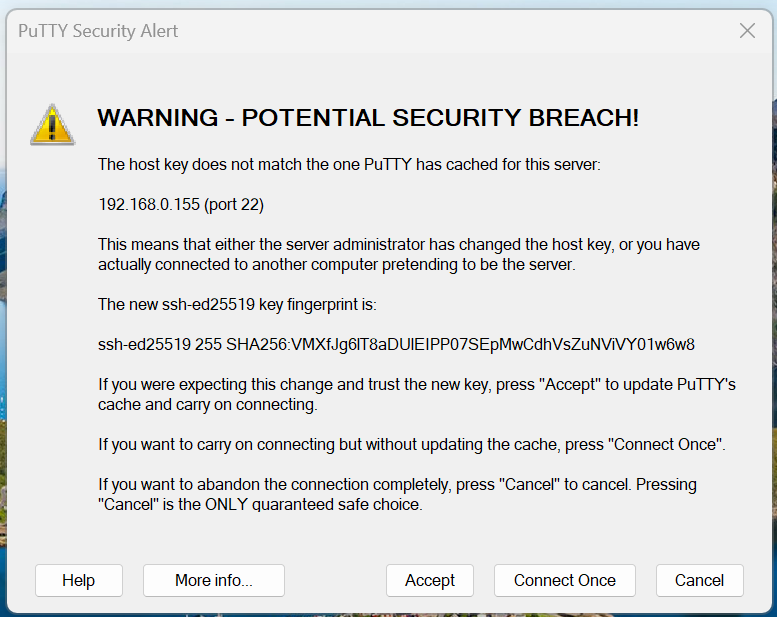
If you are familiar with another terminal program that supports “SSH” then feel free to use that.

Launch “Putty” and enter the following information…



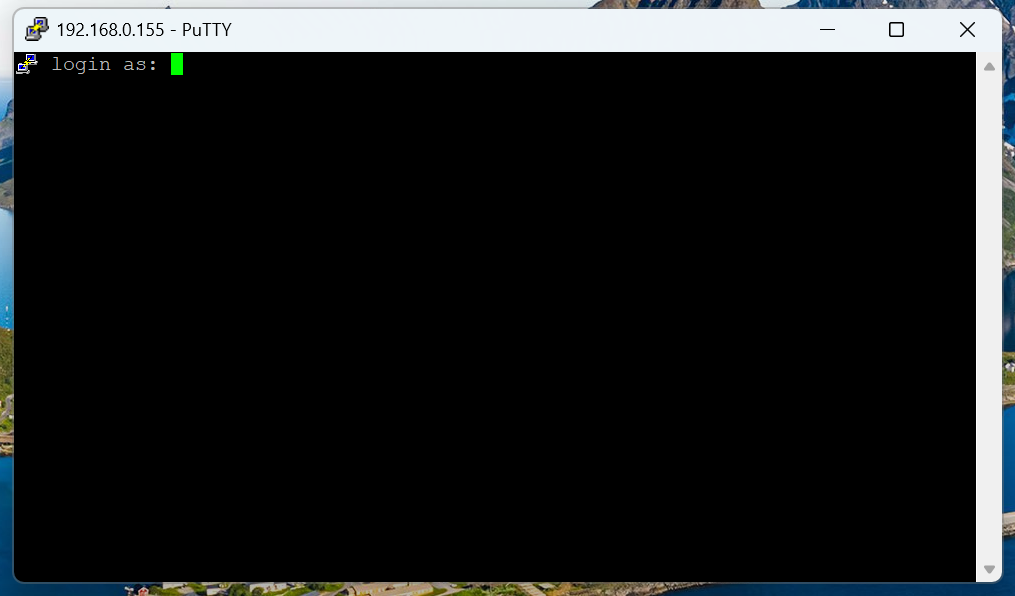
Replace ‘192.168.0.155’ with the IP that was displayed earlier when you powered up the Raspberry Pi, make sure SSH is selected and click “Open”.

The first time you connect to the station you will get the following warning.

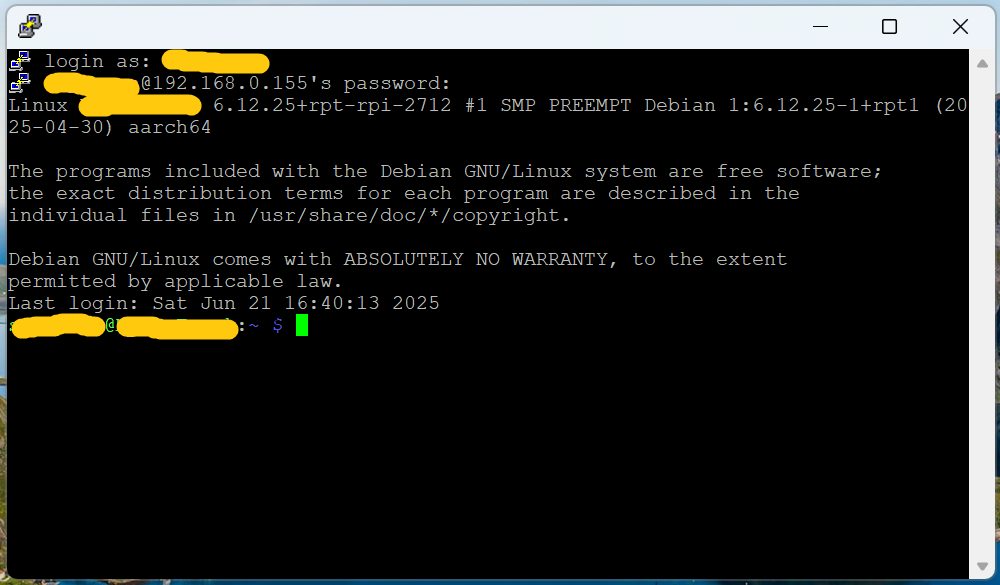


Click “Accept”.

You should now see a login screen similar to the below.



Enter the Attendance Tracker account credentials you established and you should see a screen similar to below.



First we are now going to make sure that the Raspberry Pi operating system is updated to the latest.

Type the following commands.

sudo apt update

sudo apt ugrade

Now we are going to make some changes to enable things like the real time clock (RTC) and the UART interface to the RFID interface.

Type “sudo namo /boot/firmware/config.txt” to open the configuration file for editing.

Using the cursor keys scroll down to the bottom of the file.

At the very end of the file add the following 2 lines of text.

dtparam=rtc\_bbat\_vchg=3000000

dtparam=uart0

Followed by ‘Enter’

Next, press ‘ctrl’ & ‘s” to save the file, then ‘crtl’ & ‘x’ to exit the editor.

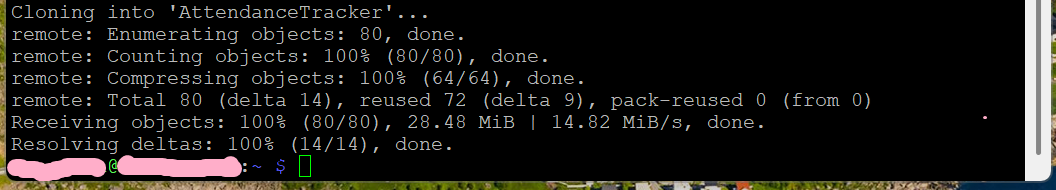
Now we are going to download the Attendance Tracker application.

Type the following commands

cd

git clone <Repository location>

You should see something like the following.



Now we need to make the program launch when the Raspberry Pi has loaded the desktop.

Type the following commands

cd

nano $HOME/.config/wayfire.ini

Once the editor loads the file move to the bottom of the file. As you scroll through the file check if there is a section the starts with the text “[autostart]”. Initially this will likely not exists, so don’t worry, but the Raspberry Pi Foundation has a habit of changing things constantly, so this information may be out-dated even as I type!!

If there is NOT a section with “[autostart]” then move to the end of the file and add “[autostart]” at the bottom.

Immediately on the next line after “[autostart]” add the following text.

Attendance = $HOME/AttendanceTracker/start.sh

Now press ‘crtl’ & ‘s’ to save your changes, then ‘ctrl’ & ‘x’ to exit the editor.

Finally type the following to reboot the Attendance Tracker and implement the changes we have made.

sudo reboot now

Note you will lose the connection we have been using in Putty, so don’t worry about the ‘Fatal error’ from Putty. Just close Putty for the moment.

Once the Raspberry Pi has rebooted the desktop should load and a short time after the Attendance Tracker should launch.

## Google Sheets interface

### Google sheets limits

#### Workbooks

10 million cells

18,278 columns

200 sheets

Update 40,000 rows at once

#### Data transfer

Read requests

300 per minute per project  
 60 per minute per user per project \* (This is effectively the limit for attendance tracker unless each station is configured as a different user)  
Write requests

300 per minute per project  
 60 per minute per user per project \* (This is effectively the limit for attendance tracker unless each station is configured as a different user)

## ToDo :

Encrypt the member name field and details

Encrypt the member photo

Create template Google Sheets workbook

Useful links

## Pi5 pinouts, UART etc…

<https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>

<https://www.youtube.com/watch?v=27p4XHE3iyw&t=451s>

<https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#real-time-clock-rtc>

<https://github.com/thagrol/Guides/blob/main/bookworm.pdf>

Python Google Sheet API demo

<https://www.youtube.com/watch?v=zCEJurLGFRk>