Document to setup the credentials.json file and to run the application.

MeetingMate - Google Calendar Automation Tool

Prerequisites - Python 3.8 or newer - Google account

Step to setup

Step 1 . Set Up a Virtual Environment**:**

python -m venv env

source env/bin/activate # On Windows use `.\env\Scripts\activate`

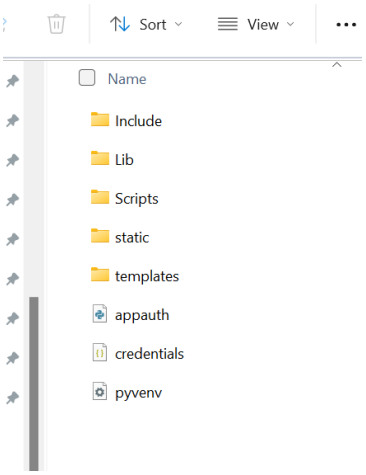
Step 2: Obtain Google API Credentials:

Go to the Google Cloud Console (https://cloud.google.com/gcp?utm_source=bing&utm_medium=cpc&utm_campaign=na-US-all-en-dr-bkws-all-all-trial-e-dr-1707554&utm_content=text-ad-none-any-DEV_c-CRE_-ADGP_Desk+%7C+BKWS+-+EXA+%7C+Txt-GCP-General+GCP-KWID_43700063334641464-kwd-76759859428126:loc-190&utm_term=KW_google%20cloud%20console-ST_google+cloud+console&&msclkid=3a28a7abf9fa1cae83d0c1fb142f98b0&utm_source=bing&utm_medium=cpc&utm_campaign=1707554%20%7C%20Google%20Cloud%20GCP%20%7C%20DR%20%7C%20ESS01%20%7C%20NA%20%7C%20US%20%7C%20en%20%7C%20Desk%20%7C%20SEM%20%7C%20BKWS%20-%20EXA%20%7C%20Txt%20%7C%20Bing_Bing&utm_term=google%20cloud%20console&utm_content=Desk%20%7C%20BKWS%20-%20EXA%20%7C%20Txt-GCP-General%20GCP&gclid=3a28a7abf9fa1cae83d0c1fb142f98b0&gclsrc=3p.ds)

Step 3:Create a new project . Further enable the Google Calendar API and Gmail API.

Then next go to “Credentials” and create OAut 2.0 credentials.

Further, download the credentials.json and place it in your project directory as shown below.



Step 4: Run the application:

Python Appauth.py

Step 5: Access the application using the below url:

http://localhost:5000/

You will also get a prompt in you command editor (as shown below for the project I have built).