

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:

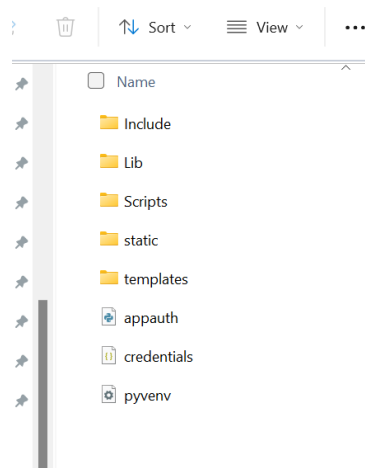
Go to the Google Cloud Console

([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](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 OAuth 2.0 credentials.

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



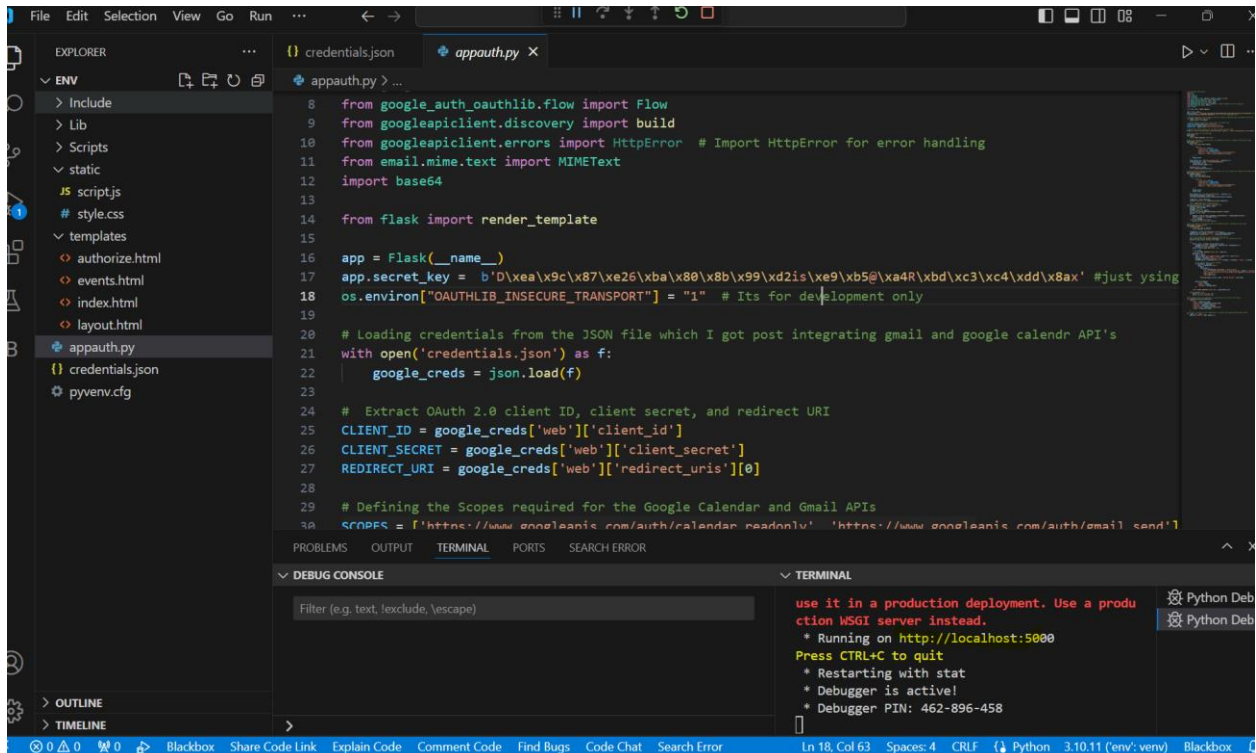
Step 4: Run the application:

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).



The screenshot shows a Visual Studio Code editor with a dark theme. The Explorer sidebar on the left shows a file structure with folders 'ENV', 'static', and 'templates', and files 'authorize.html', 'events.html', 'index.html', 'layout.html', 'appauth.py', 'credentials.json', and 'pyenv.cfg'. The main editor window displays the 'appauth.py' file with the following code:

```
8 from google_auth_oauthlib.flow import Flow
9 from googleapiclient.discovery import build
10 from googleapiclient.errors import HttpError # Import HttpError for error handling
11 from email.mime.text import MIMEText
12 import base64
13
14 from flask import render_template
15
16 app = Flask(__name__)
17 app.secret_key = b'D\xea\x9c\x87\xe26\xba\x80\x8b\x99\xd2is\xe9\xb5@\xa4R\xbd\xc3\xc4\xdd\x8ax' #just using
18 os.environ["OAUTHLIB_INSECURE_TRANSPORT"] = "1" # Its for development only
19
20 # Loading credentials from the JSON file which I got post integrating gmail and google calendr API's
21 with open('credentials.json') as f:
22     google_creds = json.load(f)
23
24 # Extract OAuth 2.0 client ID, client secret, and redirect URI
25 CLIENT_ID = google_creds['web']['client_id']
26 CLIENT_SECRET = google_creds['web']['client_secret']
27 REDIRECT_URI = google_creds['web']['redirect_uris'][0]
28
29 # Defining the Scopes required for the Google Calendar and Gmail APIs
30 SCOPES = ['https://www.googleapis.com/auth/calendar.readonly', 'https://www.googleapis.com/auth/gmail.send']
```

Below the code editor, the 'TERMINAL' panel is open, displaying the following output:

```
use it in a production deployment. Use a production WSGI server instead.
* Running on http://localhost:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 462-896-458
```