Cold Email Automation Script

Objective

This Python script automates the process of sending **personalized cold emails** to multiple recipients using data from an Excel sheet. It connects to Gmail's SMTP server and sends custom messages to each contact.

1. Import Required Libraries

```
import pandas as pd
import smtplib
from email.message import EmailMessage
```

- pandas: Used for reading data from Excel/CSV files.
- smtplib: Allows sending emails via the Gmail SMTP server.
- EmailMessage: Helps format and structure the email content.

2. Load Contact Data

```
data = pd.read_excel('contacts.xlsx')  # or pd.read_csv('contacts.csv')
```

- This reads the contact information from contacts.xlsx.
- The file must have the columns: First Name, Company Name, mailid.

You can switch to pd.read_csv() if you're using a .csv file instead.

3. Define Email Credentials

```
YOUR_EMAIL = 'your_email@gmail.com'
YOUR_PASSWORD = 'your_app_password'
```

- Replace with your Gmail address.
- Use an App Password if Two-Factor Authentication (2FA) is enabled. (Generated via your Google account.)

4. Create an SMTP Session

```
server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login(YOUR_EMAIL, YOUR_PASSWORD)
```

- Connects to Gmail's SMTP server at port 587.
- starttls() secures the connection.
- Logs in using your credentials.

5. Loop Through Each Contact and Send Email

```
for index, row in data.iterrows():
    first_name = row['First Name']
    company_name = row['Company Name']
    to_email = row['mailid']
```

- data.iterrows() goes through each row (i.e., each contact).
- Extracts:
  - first_name: Receiver's name.
  - company_name: Name of their company.
  - to_email: Email address of the contact.

6. Compose Personalized Email

```
    subject = f"Ideas for {company_name}"
    body = f"""\
Hi {first_name},

I love the work you're doing at {company_name} –
I specialize in building high-performance websites, branding, and smart AI-driven solutions for

If you're planning any upgrades, scaling, or even new digital strategies,
I'd love to share a few ideas – no obligation, just adding value.

Would you be open for a quick chat this week?

Best regards,
Madhavan A
"""
```

- Customizes the subject and body using the contact's name and company.
- Uses a **cold outreach tone**, offering value without pushing a sale.

7. Format the Email with EmailMessage

```
msg = EmailMessage()
msg['Subject'] = subject
msg['From'] = YOUR_EMAIL
msg['To'] = to_email
msg.set_content(body)
```

- Builds the message object.
- Sets subject, sender, recipient, and body content.

8. Send Email with Error Handling

```
try:
    server.send_message(msg)
    print(f"Email sent to {first_name} at {to_email}")
except Exception as e:
    print(f"Failed to send email to {to_email}: {e}")
```

- Attempts to send the message.
- If it fails (invalid email, quota limit, etc.), an error message is printed — but the script continues.

9. Close the Server Connection

```
server.quit()
```

Shuts down the connection with Gmail's SMTP server after sending all emails.

Gmail Sending Limits —

Account Type	Max Recipients Per Day
Gmail (personal account)	500
Google Workspace	2,000
Workspace Trial	500

- Sending 1 email to 10 people = 10 recipients.
- These limits include emails sent through any method (web, app, script).
- If limits are exceeded:
    - You may get blocked temporarily.
    - Gmail may return errors like 550 5.4.5 Daily sending quota exceeded.

Summary

This Python script reads a contact list from Excel and sends personalized cold emails through Gmail using SMTP. It handles errors and follows Gmail's limits, making it a good solution for small to medium outreach campaigns.