

AUTOMATED EMAIL SENDER

BY
TEAM PY 01



TECHPLEMENT

PYTHON DEVELOPER INTERNSHIP

DURATION - 1 MONTH

JUNE 2025

PROJECT DOCUMENTATION

PROJECT TITLE - Automated Email Sender.

PROJECT OVERVIEW -

The Automated Email Sender is a Python-based tool designed to send personalized bulk emails using recipient information from an Excel or CSV file. This solution is intended for small teams or organizations that require efficient and secure email distribution for events, campaigns, or notifications without relying on third-party paid platforms.

Key features include -

- Reading data from .xlsx or .csv files.
- Personalizing messages using name and custom message fields.
- Secure SMTP integration (supports Gmail and similar providers).
- Error handling and logging.
- CLI-based and ready for packaging with PyInstaller.

PROJECT TEAM -

Name - Py 01

Members -

1. **Hrituraj - Team Lead / Project Lead** (Designed the email sending workflow & architecture and prepared the final project documentation).
2. **Rashmithaa - Project Manager** (Implemented & tested the SMTP email sending functionality and designed the logging system for email send status).
3. **Jyotshna - Project Manager** (Prepared & validated input data files and tested the data loading functions).

PROJECT TECHNOLOGIES -

- Python 3.10+
- pandas – for data processing.
- openpyxl – for reading Excel files.
- smtplib – for sending emails via SMTP.
- email.mime – for message formatting.
- datetime – for log timestamps.
- os – for directory and file handling.

PROJECT SETUP -

1. Clone or Download the Repository

```
git clone https://github.com/iamhriturajsaha/AUTOMATED-EMAIL-SENDER.git  
cd AUTOMATED-EMAIL-SENDER
```

2. Install Required Libraries

```
pip install -r requirements.txt
```

3. Prepare The Data File

Save the recipient details in Data.xlsx in the format -

| Email Address | Full Name |

4. Update Sender Details

We can hardcode or enter interactively -

Sender Email - “tonychelseafan@gmail.com”

Password - “hsghdhhd89nssbs”

5. Run The Script

```
python email_sender.py
```

6. Check Logs

Logs for each email sent will be stored in -

[logs/email_log.txt](#)

PROJECT RISKS -

a) Gmail blocking SMTP due to security.

- **Mitigation Strategy** - Use app password and enable less secure app access.

b) Email marked as spam.

- **Mitigation Strategy** - Use proper formatting and avoid spam keywords in subject/body.

c) File format errors.

- **Mitigation Strategy** - Validate Excel/CSV headers strictly before processing.

d) Large recipient lists timeout.

- **Mitigation Strategy** - Add retry mechanisms and implement batch sending if needed.

PROJECT RESOURCES / REFERENCES -

- Python Official Documentation - <https://docs.python.org/3/>
- Gmail SMTP Settings Guide - <https://support.google.com/mail/answer/7126229>
- pandas Documentation - <https://pandas.pydata.org/docs/>
- openpyxl Documentation - <https://openpyxl.readthedocs.io/en/stable/>
- email.mime - <https://docs.python.org/3/library/email.examples.html>
- PyInstaller - <https://pyinstaller.org/en/stable/>
- SMTP - https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol

ADDITIONAL COMMENTS -

- The project is designed for internal and educational use.
- No external APIs or paid services are used.
- Future enhancements may include -
 - Email scheduling
 - HTML templates
 - Attachment support
 - Analytics dashboard

THANK YOU