Certificate Sender Web App – Complete Documentation

Project Overview

This Flask-based web application automates the process of sending **pre-generated certificate PDFs** to participants via email. It matches certificate files to names and emails listed in an uploaded Excel spreadsheet, attaches the appropriate certificate, and sends it with a custom thank-you message.

Folder & File Structure

🗱 Features

- Upload Excel file (.xlsx) with Name and Email columns.
- Upload pre-generated PDF certificates.
- Input a custom thank-you message via the web form.
- Automatically matches and emails certificates to participants.

🚀 Installation and Setup

- 1. Clone or download the project.
- 2. Install required packages:

```
pip install flask pandas yagmail openpyxl
```

- 3. Set up your Gmail:
 - Enable **2-step verification** in Gmail.
 - Generate an **App Password** under Google Account → Security → App Passwords.

4. Edit config.py:

EMAIL SENDER = 'your email@gmail.com'

EMAIL PASSWORD = 'your app password'

EMAIL_SUBJECT = 'Thank You for Participating'\



Web Interface (app.py + HTML form)

1. User uploads:

- An Excel file (.xlsx) with two columns: Name, Email.
- o One or more PDF certificates.
- o A custom thank-you message.

2. Backend processing:

- Excel file is read using pandas.
- Names are formatted (e.g., "Alice Smith" → alice_smith.pdf).
- o Matching certificate files are located.
- o Emails are sent using yagmail.

config.py - Configuration File

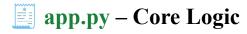
python

CopyEdit

EMAIL SENDER = 'your email@gmail.com'

EMAIL_PASSWORD = 'your_app_password'

EMAIL_SUBJECT = 'Thank You for Participating'



Key Libraries:

```
import os
import pandas as pd
import yagmail
from flask import Flask, render_template, request
from werkzeug.utils import secure filename
```

File Upload & Processing:

• Files are saved to the uploads/ directory.

Certificate filenames must match names in the Excel file after formatting:

```
formatted_name = name.strip().lower().replace(' ', '_') + '.pdf'
```

•

Sending Email:

```
yag.send(
   to=email,
   subject=EMAIL_SUBJECT,
   contents=thank_you_message,
   attachments=certificate_path
)
```

Error Handling:

• Errors like missing attachments are caught and printed but don't stop the entire process.

Excel File Format

Example Certificate Names Emails.xlsx:

Name **Email**

Alice Smith alice@gmail.com

Bob Johnson bob@example.com

reconstruction of the contract of the contract

- Certificate files must be PDF.
- Filenames should follow: "First Last" \rightarrow first_last.pdf



Run with:

python app.py

Access via: http://127.0.0.1:5000/

X Troubleshooting

- Check that certificate filenames exactly match Excel names (lowercase, underscores).
- Make sure Excel uses correct headers: Name, Email.
- If using Gmail, ensure App Password is active.
- Check console logs for missing files or email errors.

Developer Notes

- Temporary files are stored in uploads/.
- Uses yagmail for secure email via Gmail SMTP.
- Flask handles routing and rendering.
- Requires an HTML form template (index.html) with inputs:
 - o Excel file
 - o Certificate folder (as zip or individual files)
 - o Thank-you message

To-Do / Future Improvements

- Add email status feedback in UI.
- Allow dynamic certificate generation.
- Add multi-language support for emails.