Wish_Wizard Setup Instructions

This document provides a step-by-step guide to setting up and running the Wish_Wizard automated birthday email bot using Python.

1. Prerequisites:

Before you begin, ensure you have the following:

- Python 3.x installed on your system
- A Gmail account with 2-Factor Authentication enabled
- Access to a terminal or command prompt

2.Install Required Python Libraries:

Open your terminal and run the following command:

```
pip install pandas schedule plyer
```

These libraries are used for handling data, scheduling, and sending notifications.

3. Enable Gmail App Password

Since Gmail requires secure access:

1. Enable 2-Factor Authentication on your Gmail account.
2. Go to https://myaccount.google.com/apppasswords
3. Generate an App Password for “Mail” on “Windows Computer” or your desired label.
4. Copy the 16-character password. You will use this in your script.
5. Prepare the Birthday Data File

Create a CSV file named `birthday.csv` with the following format:

```
name,email,birthday
Alice,alice@example.com,1998-07-04
Bob,bob@example.com,2001-12-25
```

Make sure the `birthday` column is in the format YYYY-MM-DD.

5. Create the Main Python Script

Copy the `send_bday_emails()` function code into a Python file named Birthday_Email_Scheduler.py. This file will read the CSV, check for matching birthdays, and send the emails.

6. Modify the Credentials in the Script

In the script, replace the placeholders with your Gmail credentials:

```
EMAIL = "your_email@gmail.com"
APP_PASSWORD = "your_16_char_app_password"
```

7. Test the Script

To test:

1. Modify `birthday.csv` to include today's date as someone's birthday.
2. Run the script manually:

   ```
   python Birthday_Email_Schedule.py
   ```

3. Check your email inbox and sent folder.
4. Schedule Daily Execution (Optional)

To run the script automatically every day:

1. Create a new Python file named `birthday_scheduler.py` with the following code:

   ```
   import schedule import time from birthday_email_bot import send_bday_emails

   def job(): send_bday_emails()

   schedule.every().day.at("08:00").do(job)

   while True: schedule.run_pending() time.sleep(60)
   ```

2. Run the scheduler:

   ```
   python birthday_scheduler.py
   ```

You can change the schedule time or test it by using `schedule.every(1).minutes.do(job)`.

9. Logging

The script creates a `sent_log.txt` file that logs each sent birthday email with a timestamp for reference.

10.Notes

- Ensure the CSV file and Python scripts are in the same directory.
- Do not share your Gmail App Password.
- The script sends confirmation emails to yourself to confirm success.

This completes the setup and usage of the Wish_Wizard bot. It ensures birthdays are never forgotten and greetings are always sent on time.