Emilio Sese

Prof. Fred Annexstein

April 18th, 2022

Python Programming CS2021

Cincinnati Film Lab Automation

**Introduction:**

Cincinnati Film Lab is a small business ran that develops film photography. Being a new business based around the University of Cincinnati’s campus, the team is very small and most work is done manually by the owner. Rolling, washing, loading, the film, and converting the raw photos into a digital format, takes quite a long process for one person. The owner needed help to speed business up.

My project is a python program that utilizes excel scraping and two Google API’s: Gmail and Drive. It gathers order data from the excel file and finds the order photo folder using the Drive API. Then it sends that shared folder to the customers email address through Gmail. Processes and outline fleshed out below:

Pre-requisites:

1. Photos are manually placed within Google Drive root folder (photos repository)

* **Folder repository link** is recorded for external Gmail API
* Orders are created in their own photo folder (named as the *order #*) and placed in the root photo repository

1. Sorted excel sheet contains:**customer name**, **email**, and ***order #***

* **Manually inputted**

1. “**Ready**” box/column is manually changed and gets updated when everything is sent out (send queue - can be manually manipulated)

* *In progress*
* *Ready*
* *Done*

Running .py Program

1. Scans excel (.xlsx) file for all instances of orders checked as “*Ready*”
2. If order is checked as *‘****Ready’***, program finds row and takes instance **NAME**, **EMAIL**, and **ORDER\_#** as variables
3. Since **ORDER\_#** == name of folderin root folder

* Use Drive API to find folder with name: **ORDER\_#** using link of root folder originally created
* Change permission to access folder to “Anyone who has link” or just need to record the “*Share Link*”
  + **SHARE\_LINK** recorded as variable to be inserted in email

1. Use signature within program to create an **EMAIL\_BODY** using **NAME, EMAIL, ORDER\_#,** and **SHARE\_LINK**
2. Sends email using Gmail API
3. Go back into excel file and change **‘*Ready’*** to ***‘Done’***
4. Repeat for all orders marked as ‘***Ready’***

**Architecture Diagram:**

Diagram, schematic

Description automatically generated

**Project Results:**

*Screenshots of the testing environment*

A screenshot of a computer

Description automatically generated

* Excel (*customer-info.xlsx*) – contains all information needed to carry out objective
  + Only the rows marked with a green “has folder” should carry out
  + Other orders do not have a folder, or are not marked as ready

A screenshot of a computer

Description automatically generated

* Google Drive – root folder is shown with all photo folders
  + Same Google account is used for sending the email (parent email)
  + Folder access permission is all set to “Private” or “Not shared”

A screenshot of a computer

Description automatically generated

* Recipient email (miovsese@gmail.com) – inbox of who will be receiving emails from the parent user ([damianphoto1test@gmail.com](mailto:damianphoto1test@gmail.com))
  + Empty for test cases

*Running gmail\_test.py*

A screenshot of a computer

Description automatically generated

* Console outputs all functions that are successful and throws errors for the folders that are marked as “*Ready”* but have no Drive folder
  + Only Orders #1001, #1050, and #999 were sent out

*Actual Results*

*Table

Description automatically generated*

* Excel (*customer-info.xlsx*) – Orders that were sent out are now marked as “*Done*”

*Graphical user interface, text, application, email

Description automatically generated*

* Google Drive – order folder permissions are now changed and shared
  + Icons are changed

*Graphical user interface, text, application

Description automatically generated*

*Graphical user interface, text, application, email

Description automatically generated*

* Email – received emails, with links to each folder and link to schedule meetings on website

**Division of Work:** As far as the coding portion of the project, it was all done by myself. My partner in crime and person who presented this opportunity to me (owner of Cincinnati Film Lab) is Damian Langstaff of the Lindner College of Business. He takes care of the manual photo developing and Google Drive uploads.

**Bibliography:**

* [Python Quickstart  |  Gmail API  |  Google Developers](https://developers.google.com/gmail/api/quickstart/python)
* [Sending Email  |  Gmail API  |  Google Developers](https://developers.google.com/gmail/api/guides/sending#python)
* [Share files, folders and drives  |  Drive API  |  Google Developers](https://developers.google.com/drive/api/guides/manage-sharing)
* [Reading an Excel File in Python | WD (webdamn.com)](https://webdamn.com/reading-an-excel-file-in-python/#:~:text=Reading%20an%20Excel%20File%20in%20Python%201%20Install,to%20read%20an%20excel%20file%20and%20display%20data.)
* [Getting Started with Gmail API in Python (For Beginners) - YouTube](https://www.youtube.com/watch?v=7X3fBlMw_1k&t=383s)
* [Generate File Sharing URL With Google Drive API in Python - YouTube](https://www.youtube.com/watch?v=eO-7RIMTjOA&t=386s)

**Code Appendix:** [sesemeseeds/cfl-automation (github.com)](https://github.com/sesemeseeds/cfl-automation) (code is also included within .zip folder)