

Alumni Video Donations Final Project Report

Simone Davison, Andrew Lanum, and Austin Morin

June 1, 2022

1 Project Final Summary

As of Spring Quarter on June 1, 2022, the Alumni Video Donations project is in a fully functional state and works as intended. Our Python program can take a valid donor file in a .csv or .xlsx format and generate a hyperlink to a website with a weaved video for each donor. Although multi-processing is currently throwing lint error messages when run, these are negligible and the core functionality remains completely intact. The .csv functionality is underdeveloped compared to the .xlsx reading and needs to be updated or dropped. The Python video weaver will extract resources from our remote WWUCSGiving Firebase account, weave the extracted items together, assign an access token to them, and upload these to a finished videos folder in Firebase as a .mp4 file. Data from 2022 give day, May 26, videos were sent to 95% of non-faculty recipients, the other 5% of emails bounded, and 100% for faculty recipients. Nearly 2000 videos were made without error. The project website <https://wwucsgiving.web.app/> is hosted remotely on Firebase and has been tested for compatibility on BrowserStack. It is able to render video, audio, background music, and text successfully on a majority of browsers. The website also works on iOS, Microsoft, and Android devices and is formatted to fit those resolutions. Appending the name of a weaved video in the WWUCSGiving Firebase account to the website's URL will render it. All video content and website text is designed to our customer Dr. Filip Jagodzinski's expectations and is personalized to the corresponding recipient of the video.

A day before Give Day 2022, the website was sent to all valid recipients via email. After May 27, 2022 at 11:59pm, the Computer Science Department made \$926 more than 2021. In addition, the Computer Science Department rose from 16th in the Give Day Top Fundraisers Leaderboard to the 11th in the Give Day Top Fundraisers Leaderboard.

2 Completed Project Parts

- **Storage Access (storage_access.py)**
 - Downloads and uploads from Firebase
 - Filters files from Firebase
 - Downloads content from Firebase based on graduation year, content type (photo or video), and amount
- **Video Weaver (weave.py)**
 - Running on the command line takes in user input and parses it indexes.
 - Creates personalized videos that contain text, music, and clips and photos of a specified year.
 - Parses donor files and splits it among the specified number of processes.
 - Divides the video making function among processes, sending minimal data between the parent process and the child processes.

- Outputs an identical file to the original with links to each donor’s video in the final column.
- Website
 - Hosted on Firebase.
 - Displays the video specified in the URL.

3 Unfinished Project Parts

- Storage Access
 - Implement a way for users to see both a complete list of files on Firebase and a list of files in a specified directory via the Python console via a print message.
 - Implement an intuitive way for users to overwrite or delete files from specified folders and directories in Firebase remotely via the Python console.
- Video Weaver (weave.py)
 - Configure the program and multiprocessing to work on the cluster to speed up video weaving process. Currently runs on a single lab machine at 1.5 videos per minute, and when SSHing into a lab machine runs at 1 video per minute.
 - Solve lint errors during multiprocessing. Caused when a process tries to delete a file that it thinks it is still using. As code is written, the program should be done with the file by the time it deletes it. Files do eventually get deleted, but it can take a while, and it takes up precious space.
 - Better incorporate class and faculty photographs into the final videos while maintaining quality. `storage_access.py`’s `downloadContentFromStor` function does allow for videos to be added, and `weave.py` does incorporate photos. The photos taken were not of decent quality, so they were not used in the final project.
 - Create an executable file of the program.
 - UI, we did not touch that in our rendition of the project, but to have a pop up window that prompts for a file and column names would be easier to use than the command line interface.
 - The program can take in both CSV and XLSX files, but towards the end we stopped development on the CSV reading, parsing, and writing. Thus, it does not contain the latest
- Website
 - Re-package website so that it can be repurposed for other departments beyond Computer Science.
 - Implement a system that dynamically changes the links and QR codes in the video and website content to correspond with on-campus events (i.e. Give Day).

4 User Documentation

Both the User’s Guide and Firebase Storage guide have been submitted on the Final Project Canvas assignment page, along with this Final Project Report. Copies are also found on the master branch of the Alumni Money GitLab repository, and the project’s WWUCSGiving Firebase account.

5 Developer Notes

This program was requested by our Client Dr. Filip Jagodzinski to work on the lab machines. As the program is written, it is quite labor-intensive as the number of threads is not throttled. In `weave.py`, the maximum number of processes the program will use is the variable `maxWorkers`. On the lab machines it is advised to keep it down to 4 workers, but even so, the number of `ffmpeg` threads stack up endlessly and shut down background processes. After discussions with CS support, we are to run the program in the `tmp` folder on the lab machines to isolate the program to that one machine and not slow down the network.

Cleaning the donor files:

Eliminate any persons on the list without an email, remove businesses/corporations/non-people, and check for any duplicates in all the files given. For the graduation year column, make sure it is the graduation year from western. If it is the graduation year from anywhere else, clear the cells containing grad years from other institutions to indicate they did not graduate from western. OPTIONAL: Eliminate all unnecessary columns. For our program, it needed at least the columns first name, last name, grad year (from western). For the CS office, which sends out the emails, they need the output file to contain at a minimum, first name, email, and link to their video. Union those and get first name, last name, grad year, email, and running it through the program will produce an output file containing the video links.

Running the program:

As the program would sometimes end abruptly, we suspect due to thread overload, we opted to split up the donor file into batches of size 100, then take all the output files and merge them back together. Ideally, throttling the program's utilities usage will make that unnecessary.

For questions, feel free to reach out to the previous team:

- Simone Davison - simonedavidot@gmail.com
Developed multiprocessing, file parsing, and Firebase access in `weave.py` and `storage.access.py`
- Andrew Lanum - andyланum@gmail.com
Developed the website, and video weaving in `weave.py`
- Austin Morin - austin.morin6224@gmail.com
Finished the website, `clean_worksheets.py`, operating photo and video cameras, and advertising project