**Project Report**

1. Introduction
   1. What is the project?
   2. Why is the project needed?
   3. Project goals
2. Motivation and Background
   1. Challenges of interactivity
   2. Previous approaches
3. Corpus
   1. Reasoning
      1. Student experience
   2. Design
      1. University History
      2. City History
4. Technical Design
   1. Requirements
      1. Create calendar
      2. Event for each month
      3. Printable and usable
   2. Calendar Design
      1. Placement of image and text with qr code
      2. Size of image and text
   3. Architecture
   4. Diagram of back end to the corpus
   5. Workflow
5. Implementation & Development
   1. Technologies used
   2. Multiprocessing
      1. Huge speed increase, 3m30 down to <1m
   3. Callirhoe & Calmagick
      1. Callirhoe Designs
      2. Calmagick Stitching
   4. Event Retrieval
      1. Google docs to CSV
         1. Authentication
         2. Gspread
      2. Excel to CSV
         1. Pandas read excel
      3. CSV parsing
         1. pandas
      4. Choosing events
         1. Shared csv files between two scripts
   5. Displaying Event
      1. Downloading image
      2. Downloading text
      3. Creating QR code
   6. GUI
      1. Argument Parser
         1. Gooey Parser
      2. Gooey
         1. Using Gooey
         2. Multiple Gooey Windows
6. Evaluation
   1. Usage Testing
      1. Handing out Test calendars
      2. QR tracking with bit.ly
   2. User Feedback
      1. Design Feedback
      2. Functionality Feedback
      3. Overall Engagement
   3. Hopefully evaluation from the original blog makers
7. Conclusion
   1. Successes
   2. Failures
   3. Future Work
   4. Personal Achievements

**TESTING**

coverage3 run -m unittest tests/test\_selector.py tests/test\_parser.py

.

.

.

.

Authorizing access to spreadsheet

Opening spreadsheet

Creating CSV file from spreadsheet

.

.

.

Creating CSV file from spreadsheet

Your csv file has been saved in Events/test.csv

.

----------------------------------------------------------------------

Ran 8 tests in 1.292s

OK

--------------------------------------------

lib/\_\_init\_\_.py 60 50 17%

parser.py 42 15 64%

selector.py 75 50 33%

tests/test\_parser.py 39 1 97%

tests/test\_selector.py 28 1 96%

--------------------------------------------

TOTAL 244 117 52%