**Name: Darren Key**

**Email: darren8key@gmail.com**

**Assignment worked on: Option 1**

**Started: @ 6/1 11:10 AM**

**Ended: @ 6/2 9:40 AM**

**How long I spent - ~12 hours in total; I finished the very very minimum at about 3 hours and coded in a backend in 1 more hour, and the rest was spent making the project look better. Though after spending a lot more time on the project, I would classify the project is unfinished, and A LOT more could be added; there’s always more to add, such as adding in more stats, made the project look better, etc.**

**Instructions on how to use my project: Run the runserver.py file included at the parent directory**

**Explanations of struggles - Some slight struggles; read the logs for more information**

**Any other documentation - read the Documentation.txt file**

**CLICK on the USERNAMES of the posters on the home page(**[**http://127.0.0.1:8000/**](http://127.0.0.1:8000/)**) to get to the page of the user’s most recent posts**

**CLICK on the STATS button of the posters on the home page(**[**http://127.0.0.1:8000/**](http://127.0.0.1:8000/)**) to get to the page of the user’s stats**

**Also included is my favorite photo ☺**

**You can also see previous iterations of my webpage in the Previous iterations folder (but the formatting for v2 is messed up)**

Logs of my process:

Initial thoughts:

I've chosen to use Django due to my familiarity with Python. Django can handle both assignments (and once the data from the csv file is imported to the models both assignments should be equally doable), so I've just decided to go with option 1.

The two biggest challenges I see so far are:

a) Reading the csv file and importing them to Django's models

b) Especially parsing the JSON part of the csv file

-- 11:42 AM--

Finished setting up the basic elements of the Django server. I ran into a couple of hiccups (my Django was outdated), but it was easily fixable. I mainly utilized the fields documentation for this step (https://docs.djangoproject.com/en/3.2/ref/models/fields/#django.db.models.Field). There was a nice field to decode the content portion of the CSV, called the JSONField (which wasn't available in Django 3.0, which was the version I used).

My goal now is to import the data to the models I've created.

--12:16--

I think the simplest (and fastest way) for me currently is to just use the built-in Command library given in Django. For this, I'll use the csvreader (https://docs.python.org/3/library/csv.html) to parse the csv file.

I've ran into a little issue where the fields of the csv file are too big for csv to handle. This is an easy issue to deal with; thanks to Stackoverflow I can just increase the max\_limit of the csvreader.

--12:41--

I used the Command methods in Django to import the csv data. Although this way is acceptable (I learned this way), in hindsight, there are faster and better ways to do it; searching it up I've found several libraries created with the ability to auto import csv data, including the import-export library and the csvimport library. There also exists ways to do it without coding, by manually using a program to import data into the .sqlite file.

Now, I need to parse the JSON of the content section.

--12:50 - 1:08--

Lunch

-- 1:49 PM --

A little update:

I've been able to parse the JSON (fairly easy, I just imported json and used json.load(), which gives a good enough response). What I'm doing now is just going through the JSON and finding the elements which I would like to display on the website.

--2:20 PM--

I finished the absolute BARE MINIMUM of the project. In total, it took me about 3 hours. The webpage for Option 1 right now is very, very, very bare bones and just displays each user's username, profile picture, profile url, and description.

It's included in the current project as ChartmetricVersion1.html. I'm definitely going to beautify and spice up the webpage to make it better.

Here's the log in a list if needed:

Assignment worked on - Option 1

Time spent - 3 hours

Struggles with assignment - No major struggles; the only parts which were a little tricky to deal with was importing the data to Django (I spent about an hour trying to look for "better" alternatives to the method I knew worked) and parsing the JSON (since it has specific keys and a specific format).

Other notes -

The beauty of Django is that now I can REALLY easily do option 2, as well as import any more data if needed really easily through my code. I can also easily add data through Django's built-in admin page. Since I have the time (and the time in the assignment; I've only used about 3 hours of the 24 hours), I'm going to go ahead and do Option 2 as well and beautify and cleanup Option 1 (because why not and it looks like fun :) ).

---Short 45 min break----

--3:22 PM--

Got back from break, there was a slight bug in the webpage (the profile links didn't work) so I fixed it pretty easily. Also realized I missed out on a lot of information in the content JSON, so I'll fix that up. In the meantime though, I think I can do Option 2 pretty easily using Django here.

--4:33 PM--

Coded Option 2 just for fun. Documentation included in the Documentation.txt file. Now I'll focus on making the initial webpage better. I realized that I had A LOT missing here, so I'll be focusing mainly on here from now.

--9:10 PM--

After several breaks, dinner, and more coding, the webpage looks presentable now. ChartmetricVersion2.html is my latest version. I'll continue to add more though and will probably add several more pages to the website to fully flesh out everything.

--9:54 PM--

Adding a page with recent posts and top posts. Had a slight struggle with making the users clickable but after some testing and much stackoverflow, I made the usernames clickable through some inline javascript.

--11:40 PM--

Finished adding recent posts and top posts and made everything neat. My focus now is to add stats and charts and everything.

--9:37 AM--

Added in some stats. A lot can be added to my webpage, but my time is up.