This assignment was locked Oct 11 at 9am.

Corrections

* Before class Thursday 9/29:
  + Extended 2nd check-in deadline to Tu 10/4 at 11:59 PM
* Before class Tuesday 9/20:
  + Corrected "student object" to "user/profile object" in the grading rubric
  + Fixed "The first non-whitespace characters" part of the spec for the JSON file to match the actual sample file! [See this Piazza thread.Links to an external site.](https://piazza.com/class/l7888407g0b6by/post/28)
* After class Thursday 9/15:
  + Deadline changed to Sunday, October 9 (which, unlike October 8, is actually a Sunday)
  + Dates for second Piazza check-in changed
  + Added note about modifying the input generator
  + Fixed the description of the Social Network class under “Pro Tips”
  + Added reminder about the collaboration policy

Project Overview and Deadlines

In this project, you will read data describing the structure of a social network from an input file, build data structures representing individual profiles and the overall social graph, and output a set of HTML files representing profile pages and the overall network.

This project is due at 11:59 PM on Sunday, October 9.

There are two weekly check-ins on Piazza. The first one needs to be posted between 12:01 AM on Tuesday 9/20 and 11:59 PM on Thursday 9/22. **UPDATED:**The second needs to be posted between 9:00 AM on Thursday 9/29 and 11:59 PM on Tuesday 10/4.

To receive a style grade on this project, you will need to make a 1-on-1 appointment with me between 10/10 and the beginning of class 10/25.

Input Description

For your final submitted code, assume that your input file is named *input.json* and lives in the same directory as your executable.

Your input file will be a JSON file containing an array of users.Each user is a JSON object with the following attributes, which may be listed in any order:

* id\_str: an integer represented as a quoted string. Every user is guaranteed to have an id\_str. If you have N users, they will have id\_str values of “1” through “N”, but they will not necessarily appear in the file in that order.
* name: a quoted string. Every user is guaranteed to have a name.
* location: a quoted string. A user may or may not have this attribute.
* pic\_url: a URL, as a quoted string. A user may or may not have this attribute.
* follows: a JSON array of integers represented as quoted strings. This is a list of the id\_str values of the people this user follows. This array may be empty. If it’s not present at all, you should infer that it’s empty and that this user does not follow anyone.

Whitespace should be disregarded, except in quoted strings. You cannot count on having a guaranteed amount of whitespace between tokens.

You can expect that:

* The first non-whitespace characters in the file will be

{“users”:[

* The last non-whitespace characters will be

]}

* There will be at least one user.
* Users will not have any attributes other than those listed above.
* You don’t have to worry about escaping quoted strings – in other words, nobody’s name, location, or URL will contain a double quotation mark.

Sample Input

I’ve created a sample input file with 4 users: [sample-input-4.json](https://canvas.sonoma.edu/courses/33159/files/2916521?wrap=1)[Download sample-input-4.json](https://canvas.sonoma.edu/courses/33159/files/2916521/download?download_frd=1)

I’ve also created a tool to let you generate arbitrarily large sample inputs. You should grab the two files:

* [input\_gen.py](https://canvas.sonoma.edu/courses/33159/files/2916519?wrap=1)[Download input\_gen.py](https://canvas.sonoma.edu/courses/33159/files/2916519/download?download_frd=1)
* [names2021.py](https://canvas.sonoma.edu/courses/33159/files/2916520?wrap=1)[Download names2021.py](https://canvas.sonoma.edu/courses/33159/files/2916520/download?download_frd=1)

Put them in the same directory on any system that has Python 3 installed (including blue). The parameters you may want to adjust are at the beginning of input\_gen.py:

* NUM\_USERS: the number of users!
* PROBABILITY\_OF\_FOLLOW: mess with this to change how dense or sparse the follower matrix is – in other words, how likely these users are to follow each other.

Feel free to go nuts adding more locations or profile pics too.

If you modify the input generator or develop one of your own, feel free to post that code to Piazza -- it's not project code, so it's totally fine. If you'd like to request modifications, post those requests to Piazza too! I'll get to them as soon as I can, and if someone else gets there before me, that's great too.

To generate an input file, run input\_gen.py. If you’re at a command line, you can redirect the output to a file by doing something like:

python3 input\_gen.py > testInput.json

Output Description

You need to create a suite of HTML files: one index.html file, and then one profile html file for each user. Here’s an example of the HTML files you’d generate for the 4-user JSON file above:

* [index.html](https://canvas.sonoma.edu/courses/33159/files/2916513?wrap=1)

[Actions](https://canvas.sonoma.edu/courses/33159/assignments/325662?module_item_id=1539580)

* [user1.html](https://canvas.sonoma.edu/courses/33159/files/2916514?wrap=1)

[Actions](https://canvas.sonoma.edu/courses/33159/assignments/325662?module_item_id=1539580)

* [user2.html](https://canvas.sonoma.edu/courses/33159/files/2916515?wrap=1)

[Actions](https://canvas.sonoma.edu/courses/33159/assignments/325662?module_item_id=1539580)

* [user3.html](https://canvas.sonoma.edu/courses/33159/files/2916516?wrap=1)

[Actions](https://canvas.sonoma.edu/courses/33159/assignments/325662?module_item_id=1539580)

* [user4.html](https://canvas.sonoma.edu/courses/33159/files/2916517?wrap=1)

[Actions](https://canvas.sonoma.edu/courses/33159/assignments/325662?module_item_id=1539580)

HTML Index File

Open the example in a text editor or choose View Source… to see the actual HTML. Here are the requirements:

* It needs to be a valid HTML file, with the <!DOCTYPE html> <html> at the beginning and the </html> at the end
* It needs to have a header with a title. You can make the title something more inventive than the “My Social Network” in the example.
* It needs to have a body section with an <h1> heading. Again, yours can be more creative than mine.
* Finally, the body needs an ordered list that lists each user’s *name* and links to their profile. This should be in order by their *id\_str*.

Any fancy formatting, etc., is up to you.

Profile Page

A user’s profile page should be named *user[id\_str].html*. For example, if the user’s *id\_str* is “1”, the profile page would be user1.html.

Here are the requirements for the file contents:

* As with the index file, it needs to be a valid HTML file, with the <!DOCTYPE html> <html> at the beginning and the </html> at the end.
* It needs to have a header with a title, and that title should contain the user’s *name* attribute and the word “Profile”.
* It needs to have a body section with an <h1> heading that includes the user’s *name* and (if present) *location.*
* It needs to include the profile pic as an image, if the *pic\_url* attribute is present.
* It needs to have a section for the people this user follows, starting with an <h2> heading and then containing an unordered list of the people the user follows. As with the index.html site, these users should be listed by name, and that name should link to their profile page(s). If the user doesn’t follow anyone, instead of an ordered list, you should have something like <p>None</p>.
* It needs to have another section for the people who follow the user, with its own <h2> heading and list.
* Finally, it needs to have a third section for Mutuals – people who this user follows, and who follow this user back. This section should be structured like the previous two.

Submission

Please submit your code, INCLUDING a makefile that compiles it on blue, as a .zip file on Canvas. You may also submit sample input files.

Correctness Grading

Here’s a general breakdown of how you’ll be graded. If you don’t have the project fully working, be sure that your main() function shows off what you *are* able to do!

|  |  |
| --- | --- |
| Parse a single user/profile object and emit an HTML file with name, location, and picture (if applicable) | 30 points |
| Parse an array of user/profile objects and emit an index.html file linking to profile HTML files with name, location, and picture (if applicable) | 30 points |
| Include information about follows, followers, and mutuals in the individual profile pages | 30 points |
| Post to the weekly check-in on Piazza: what have you done since the last checkpoint, and what are you still working on? | 5 points each |

Style Rubric

We’ll also go over your programming style, and you’ll get a practice grade for that out of 10 points. Here’s what I’m looking for:

|  |  |
| --- | --- |
| Documentation: Do you have a comment at the top of each file with your name and a description of what it does? Do you have a comment at the top of each function? Do you explain anything tricky in your code? | 3 points |
| Names of variables, functions, classes, etc: Are they clear and precise? | 2 points |
| Decomposition into classes and functions: are the interfaces clear? Are you moving repetitive code into functions? | 3 points |
| Formatting: Is your formatting consistent? | 2 points |

Pro Tips

I **strongly**recommend creating the following data structures:

* A User or Profile class that contains the profile attributes for a given user and is responsible for outputting individual profile pages
* A SocialNetwork class that manages the entire network, including:
  + A matrix (2D array) of bools that keeps track of follower relationships and is responsible for outputting the index.html file.
  + A vector or other linear data structure that stores all the User/Profile objects
  + A member function that emits the index.html file and calls the per-user functions to emit the profile HTML files.

Collaboration Policy

This project is *solo*work, though I encourage collaboration, especially on Piazza, on everything but the actual code. See the syllabus for the full project collaboration policy.

Rubric

**Project 1 Rubric**

| Project 1 Rubric | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a Learning OutcomeIndividual User |  | 30 pts |
| This criterion is linked to a Learning OutcomeArray of Users |  | 30 pts |
| This criterion is linked to a Learning OutcomeFollow(er)s + Mutuals |  | 30 pts |
| This criterion is linked to a Learning OutcomeCheck-In 1 |  | 5 pts |
| This criterion is linked to a Learning OutcomeCheck-In 2 |  | 5 pts |
| Total Points: 100 | | |

[Previous](https://canvas.sonoma.edu/courses/33159/modules/items/1548192)[Next](https://canvas.sonoma.edu/courses/33159/modules/items/1548162)