Below is an abridged summary of my development process and some of the issues I ran into when trying to build this web application. Aside from Google, I utilized three different Slack teams to get help with this project (Cincy Tech Slack #python; Cincy WWC #dev-questions; Code Newbie #python).

Repository Issue Listing App Development Process:

Step 1: Get a conceptual idea of how the different pieces of the app connect together.

My previous experience had involved building web apps that pulled from databases like MySQL that sometimes I'd simply populate myself. I wanted to figure out where the "Python SDK" and everything else fit into this picture.

With the help of a couple of people in the Cincy Women Who Code Slack, I was able to get a conceptual idea of what I was trying to build.

Step 2: Choosing a framework.

Prior to our coffee interview, I was attempting to make a blog app and I asked for recommendations on what to use in a Slack team for women in tech. Many people felt Flask was the easiest and best framework for small projects. Based on this information, I chose Flask. [For the blog, I had opted to use a Django tutorial because the specific tutorial had a very good reputation.]

Step 3: Choosing an SDK.

I started with PyGithub at the recommendation of someone in the #python channel in the Cincy Tech Slack. I struggled with this and I wasn't finding helpful documentation. I ended up trying a few others, but eventually came back to it. I had much better luck the second time and this was because I just directly looked at the source code for the repository class instead of trying to find documentation.

Step 4: Figuring out how to list issues for a repo in Python using the SDK (no web app at this point).

I started with the example in the introduction in the documentation where I listed all the repos that I have. However, I found it concerning that my Github username and password were shared in my Python file so I tried to find a way around it.

After that, I looked in the repository class and experimented with various functions that related to issues. I eventually just made a list in my notebook of the methods that seemed potentially related to what I was trying to do.

Soon enough, I was getting errors and thought it was an issue with my use of the functions. However, I researched the issue ("API rate limit exceeded") and put my Github username and password back in the file. I decided I would just remove these right before I submitted the project on Github.

After this, I had some simple Python code that would print the issues for a given repo (not at the web app stage yet).

Step 5: Research if user/repository format of entering a repo would work in Python (versus needing to parse the string and find the user and then the repository).

Step 6: Since I thought that filtering for the top 10 issues would be relatively simple, I moved on and started trying to build the web app.

I got stuck when I was trying to get the results from the Python code to display in HTML. I kept doing web searches that had the words “rendering” in them and was not having much luck.

I asked about this in the Code Newbie Slack and the Cincy Women Who Code Slack and eventually people in both Slacks responded and mentioned the word “templating.” This was exactly what I needed and I was able to display a hard coded string (in Python) on my HTML page.

Step 7: Next, I needed to figure out how to use the data from the form in HTML. To keep it simple, I just tried to get the app to take the name submitted to the form and display it on the page.

Step 8: Next, I wanted to actually use the data from the form fields in functions. I got stuck here and was getting an assertion error. I discussed this problem in the Code Newbie Slack, but I still couldn’t fix it. I also shared this issue in the Cincy Tech Slack.

I later realized that part of the specific issues I was having was caused by an indentation error. Some of my code that only should have run if request.method == 'POST' was outside of the if block and that was causing some problems.

Step 9: Filter to only display the first 10 issues.