INTERVIEW QUESTIONS RESPONSE

I wanted to participate in IronHacks because I didn't have any programming classes. Programming is one of my personal hobbies, but sadly, I am not in any classes for it, and I don't have the time to fit in a computer science minor. IronHacks seemed like the perfect opportunity to keep improving my programming knowledge, solve a fun problem (it is often hard to design briefs for myself), and, as a bonus, to get an extra honors credit.

During the training, I focused on the skills that were gaps in my knowledge. I had a good amount of experience in HTML and CSS, but I had never worked with D3 and my Javascript knowledge was fairly minimal. I tried to use the IronHacks tutorials, but I found them to be very basic and not very applicable or useful. I went to the first tech session, but again, I found it too be too simple for my experience level. That being said, I did use the Codecademy JQuery tutorial that was linked, and I found that helped to form a basic understanding of JQuery that helped me in the hack. As for other videos or resources, I used random resources. I didn't know how to parse through JSON, so I used a wide range of articles and videos. There was not a singular source that I used. Stack Overflow and YouTube were the primary mediums to learn. It was hard to know what I wanted to learn before the hack because I didn't know the problem, so it is difficult to do further research. I don't want to learn aimless information that won't help me, so I did a lot of my learning after the hack started.

During the first phase, I read through the task multiple times and looked at the allowed databases and libraries to determine what might become a useful feature later on. My common workflow starts by developing the front-end user interface first then making the back-end work. This means that during the first phase, I created UI mockups in Sketch to get a rough idea of what I wanted my final application to look like. After I got the mockups, I started to turn it into code. I had a rough idea of how to make it into code, so that was what I did during the final two days of phase one before the submission. I didn't have too many issues during the first phase. I was fairly optimistic that I could implement everything that I wanted to include. I found the tasks pretty simple to understand, and I could have easily implemented all of them if the datasets gave me better data or if I was allowed to use other libraries, but I wanted to stick to the constraints as much as possible. I worked where was convenient for me, everywhere from my room to the Honors College Great Hall and Krach. I basically just worked around my other classes when I had time and wherever I was close to. I was not very picky about that. Once I got the front end coded in HTML and CSS, I was ready to submit my first commit. It was what I wanted to accomplish for the first phase, so once I was done, I just commited my code. When I first started my project, I didn't really seek any help for technical stuff. I

may have asked friends for ideas at one point or another, but nothing too crazy. My biggest constraint was the set of libraries and APIS. I understand why there needs to be an equal ground for all of the contestants, but if all of the databases were allowed, I think that you could see a lot cooler applications and mashups. Also, there were some libraries, like noUiSlider, because they could have added the polishing touches to my application and make the UI more understandable. Throughout phase one, I did not really use the tutorials. As previously mentioned, I used numerous other tutorials to help with any coding issues that I encountered. During phase one, I didn't really ask questions in the forum, but I tried as best as I could to answer the questions of others because I knew that I had knowledge that I thought could be helpful. I was happy with the first results of the competition. I was disappointed that we didn't get more detailed feedback though. If you are judging a program based off of errors, it is nice, as a programmer, to be told what those errors are in order to fix them in the next commit. I looked through every application to see what other people were working on. I found that fun and insightful. During the first phase, I only looked at the application and not the code, but I looked at how people coded in later phases to inspire some of my features. Through the later phases, I found the higher ranked ones more useful, but I often looked at the others just out of curiosity.

I have no idea which phase I most improved in. I gradually added features, and I had of plan that I prioritized to implement them. I started by getting the data, then displaying the data, then making the data look presentable. I didn't change my work ethic or my goals. I had created a plan from the beginning, and I continued to stick to that plan with minor changes as I went on. My list was prioritized based off of the task requirements, but other than that, I just worked to slowly implement the features throughout the phases.

I would not have done a ton differently if I had more time or had gotten to start over. I would change the UI a little bit to make it cleaner, and I would probably finish implementing the housing data. I ended up displaying it, but not on the map, so I would have like to have made that data more interactive and accessible to the user. I have never done another hack before, so I have nothing to compare IronHacks to. I know the time format was different, but I assume that a lot of the elements would be the same or similar. I was not planning to keep working on the application because I am unsure if it is actually a problem that needs solved. I have other side projects that I want to work on as well.

Overall, I think it was a very unique experience, but I wish the constraints allowed for a bit more flexibility and therefore creativity. I think IronHacks is a great idea, but the execution needs cleaned up a little bit.