1/16

Today was the first time being able to examine the code. I spent part of the evening looking over the previous iterations work. The hardware portion that primarily is running the Raspberry Pi is written mostly in C code while the GPS mapping system being in javascript. The way they implemented the server was interesting as well; they used node.js to initialize the server locally. I myself looked into other options such as google cloud storage for a more secure database. The previous team used their own local machines to run their server which may be a consideration for my iteration since it's a cheap way to run the server. However, I will have to consider using my roommates spare desktop since I use my laptop on a daily basis, thus interrupting the server loading in new data while the hardware is running gathering data within the city.

My next goal is to assemble the hardware and subscribe to the google API to be able to use all of the map features. Once I have the API working, I can start rearranging and enhancing the software. Note: I should read their journal entries as well to get an idea of how they came up with the original design to see if my ideas are capable of adapting to the new iteration.

1/19-1/20

Over the weekend I finally got to sit down and look at the hardware, which is a tinker board by ASUS which is similar to a Raspberry Pi but less friendly for users new to linux. WIth it is a micro-SD card that has the software and OS on it, a GPS detection unit, power cord and several adapters for display purposes. The tinker board is ran by a Debian 9 Linux OS which I'm not all too familiar with but I do know most linux commands. The first thing that happens when the device is turned in is check to see if the GPS port is plugged in and will immediately start scanning the nearby area and detect the signals it's supposed to. Currently I tried to install a form of Remote Desktop to it which was VNC. I'm still trying to figure out how to properly remote into it without connecting it to the HDMI port; however it isn't required but handy.

Now that I have an idea of how the board operates when powered on, my advisor Guy will be 3D printing an encasement for the components to ensure it's safe when out in the work trucks.

As far as I'm aware of the general setup is this:

The hardware standalone will detect any form of signal that it's trying to pick up, save it, and upload the data when able to (still trying to figure out this for safe data transfer.) Then, when it's able to transfer the data, it will transfer it to the web server that it's connected to which is powered by node.js, hosted by the programmers computer. Once the data is on the server, it will be displayed through the heatmap.js html site through a json file.

Improvements can be made, but will probably remain the same until the final months for it can be used by more than one person instead of 1-2 during the testing stages.

1/21-24

Over the past few days between work and class I decided to look at the scripts programmed for the board. I noticed that the default port numbers and IP addresses were from the past groups, so I've been trying to update it to reflect my own. I spent a majority of my time trying to boot the drive using my IP address to connect to the server, however it constantly kept rebooting after it failed repeatedly. After I find the right IP address (which I've tried 3 separate ones), it should be functional. Since the code has several ports with varying IP addresses for different group members from before, I have to track down which ones to switch. Hypothetically the device will work when booted up, but the data it uploads will try to upload to the past server still which is nonexistent. I need to update the current software to the board for it will be able to be uploaded to the correct server which is my local machine.

1/25-27

On Friday, Guy had me try a Raspberry Pi instead of the Asus Tinkerboard since the tinker board actually has had known issues with our AT&T router due to the firewall security settings. I noticed that the previous group used their phones as mobile hotspots so I might also want to try that as well. The current issue I'm having with the tinkerboard is that gqrx will refused to connect to my local IP address, which is most likely because of our ISP.

Over the weekend I booted NOOBS to the raspberry pi for a clean install. Once it was setup I transferred all of the relevant work files to the new system. Meanwhile I was able to install and use VNC on it so I can now use remote desktop to connect to it anytime from my computer instead of being forced to only work on it at home hooked up via hmdi cables.

Everything is pretty convenient now, however there are some obstacles at the moment. I have to reprogram the scripts the previous group did, such as the auto start script at boot up to start gqrx and the RF driver c file. Since I'm not very proficient at linux based scripting I've had to research how to do so for a rasperrian based linux system. I believe I still need to write permission files in the root file to allow certain permissions (which I've had to do just to be able to run the programs.)

If I cannot solve this issue, I might go back to the asus board since I still am not getting any work done.

1/28

I hit a wall with the raspberry pi since there is still a lot of software, scripts, and services I need to add to make it similar to the previous ASUS tinkerboard. On the tinkerboard, the main problem I have been having is that the IP address 127.0.0.1 for telnet will not work in command line since the connection is refused, and the same for any others that it *could* connect to such as my own ip address. The 127.x.x.x is a local domain ip so I should investigate further and test with more ip's since it can't properly update the data. I also noticed when experimenting with different telnet commands that the default ip address it pings is the past group members so maybe telnet is trying to connect to the members computers first, then the 127 domain name. The firewall may be just fine since I also tried my phones hotspot instead of my at&t router. I'm so close to getting it to function to my setup!

I also tried to get get several versions of vnc to run on the tinkerboard, however I have learned that the tinkerboard does not support remote desktop programs except through command line based ones such as putty. So if I plan on working on the tinkerboard, I must always be present with it to work on the hardware (which isn't bad once I start working on the software) but the raspberry pi works great with VNC but does not have the correct setup that was given to me like the asus board (preset scripts, permissions, root services, libraries I'm unaware of, etc.) Tomorrow I plan on investigating the telnet commands and setting up the correct ip addresses and reaching out to the past members to see if they ran into the same problems as me. .

1/29

I didn't get to do much with the tinker board today, but I did try and remote in to work on it however the connection failed since it went to sleep 15 minutes after I left home. After searching for more efficient ways for remote connections, apparently the asus tinker board does not support a visual platform type of remote connectivity, only through putty with the terminal so it's unwise to work from home with it since I need a visual aid.

One of the original group members actually did reach out to me today and gave me some advice on how to get it to boot up properly and send images of my errors. I sent it to him late at night and did not get a response until the next day.

1/30

Today I got a message from Brock Duerstock from the first group who specialized in the software engineering part of the project and admitted there were a few things who should have mentioned in the software instructions to make it more helpful for someone like me trying to read what they had done in the past. He noticed that I wasn't using the rtf-sdr usb port that enhances the signal strength and accuracy of the signals. I made

this aware to Guy since I was missing a few parts for it and he happened to have extra parts back at his office and will give them to me next time we meet.
Brock also noticed that the server the tinker board tries to connect to through telnet keeps disconnecting and could be because the server is not enabled and can be done so through gqrx which hasn't been working. Currently I'm trying to learn more about the telnet commands to get that server running to ping it continuously.

1/31
Since I cannot connect the RFFD to my local network for some reason, I decided to delve more into what could be more of the underlying issues that I could also be having; tonight I looked into the telnet and gqrx program. I ran apt-get update installs on both of them incase I was missing any other libraries or they needed updated. I discovered that the gqrx program was missing many additional components. The reason why I did the gqrx update is because I noticed in the previous groups presentations that they had a "waterfall" graph that was able to detect the spikes in amplitude while mine didn't. The telnet I was unable to resolve since it was more of a network connection error than an internal error so I focused the rest on gqrx. I then looked more closely in the last groups videos and found how they configured their gqrx settings and after a while of tinkering I managed to get a waterfall graph! Not only that, I found out how to enable the audio for I can test the surround radio waves. I even picked up on the Indianapolis radio stations such as the Classic Rock and Alternative stations at the 094.700 Mhz and 103.300 Mhz ranges. Guy gave me some new equipment as well, one being a larger antenna and adapters for I can plug it into the circuit board. I tested the radio waves with and without the smaller and larger antenna while also trying to see if there was a difference with the SDR-RTL usb adapter that can be used to strengthen the signal making it more accurate. The larger the antenna with the adapter, the further range I can go with a "cleaner" sound. With a smaller antenna, the radius lessens but the signal is slightly cleaner. Without the adapter, the sounds were a lot more scratchy but had less range. Now that I know gqrx is up and running just fine, I need to find a network which will allow me to test gqrx and the hardware so tomorrow I plan on going on campus to see if the network will allow it; I have a strong suspicion it will since the last group spent a lot of time testing it on campus.

2/1
Today I had to present my proposal; It was pretty long winded so I had to cut it short since I only had a 5 minute limit. Anyways,I searched on campus for a monitor or tv that I can plug the RFFD into for a display. I found a place at my work that I was able to use just fine and I started to run my tests. The RFFD is linux based so IU secure required a certificate just to use the wifi so instead I used mobile hotspot with my android. My

laptop was connected to IU secure, while the SERVER_IP address was target at my laptop while on my phones hotspot. I started up the script and it was able to connect! I finally got to see how the program communicated with one another. Since the RFFD has been collecting packets from my previous tests, it immediately sent the packets to the server while updating it to the local data.json file with the live data. At the same time, I saw the heatmap constantly updating. Unfortunately, since I was sitting in one place going at 0 mph, the signal caught my current location more than enough times that the area was highlighted in red (meaning it's severe) but the device will be upgraded to only operate while moving or not in the same position. The lab closed at 9, so I had to leave promptly after about an hour of testing the live model. Now I know it's a home network issue and nothing wrong with the software itself so I should configure my firewall at home more accordingly. This weekend I planned on going to my hometown as well so I can test it away to see how the firewall reacts.

2/2
At my mother's house I decided to test the RFFD within this network. She has a netgear modem with comcast as her ISP. I tested the RFFD on her netgear network and surprisingly it worked just fine. I observed the same things as I did on campus and looked more into how the data is written and saved. The server.c file is the one that writes to the data file and purposely formats it for the json format by use of printf statements. Again, since I was in one spot for a prolonged period of time, my location on the heatmap was bright red. While I was here, I also testing her other router instead of the netgear modem (which is comcast) to see if it to would work. It did, so I have idea; at my apartment we have three actual networks: Netgear50, Netgear50-G, and ATTXXXXX (where x are numbers.) Maybe the ATT router has less security settings? Tomorrow I will test the device at my dads house where he has a 10 year old ATT router without a modem to see if it too works.

2/3
At my dad's the results were nearly identical to my mothers as well! Same heatspot, not issues with servers, completely stable. Since the data I have been receiving is necessarily accurate nor good, I kept wiping it by loading in the original data given to me from downtown Indianapolis. Over the weekend I learned a few things about how the device and servers operate: If sitting in one position, the data will be highly inaccurate, which explains the reasoning why one of the requirements is to have the RFFD read signals only while in motion. Second, there is definitely a firewall issue happening where the RFFD is unable to reach the server and perform a handshake with it to read in the data. Third, when observing through gqrx, it seems… "overly sensitive" with detecting spikes. Even when I did have a lightbulb in the same room and purposely short circuit it,

it seemed as if it was unable to pick it up. Instead, it constantly kept sending packets which means it keeps sensing spikes every other second instead of only when it picks up a spike. I may have to go in a modify the strength/sensitivity of the signal weight for it doesn't pick up on them so easily, but at the same time it may differ from different antennas?
So now I plan on trying to get my apartments network to connect to the server and continue to modify the sensitivity.

2/4
Tonight I decided to connect to the ATT router instead of our Netgear router since I set up port forwarding on the ATT in the past. Finally, at long last, it was able to connect to the server and upload the packets. Once again though, it was too sensitive and kept picking up on every little amplitude. I ran the same tests and diagnostics as I did at home, such as observing the heatmap rapidly growing red on top of me. Now that I know I can safely work on it at my apartment, my plan is to collect more data with the IPL trucks while combing through the code seeing why the signal is so sensitive in the first place.

2/5
Since I had to work most of the day, I observed more of the heatmap html program and arranged how and what I need to focus on what I start working on the front end. I still need to get a google API key to run the html side effectively but luckily all I have right now is just a map layout. However I looked up how to implement a search bar feature for the google map api and realized that I do need to get the API key in order to use the search bar feature. I ended up being able to implement the search bar but it was unusable since my code was not yet given to me (I realized the key has a $300 credit each month where it's $300 free to use data but if you go over, you have to start paying). Since I'm using it personally for now, the usage is very minimal and not intensive on the google api. All I have to do now is enable the key and fix the placement of the search bar and that requirement is done! I still need to upgrade the software… So after work I looked into how the the signal weight is calculated, alongside the dbhz sensitivity. This is the setting that detects the spike in amplitude. I will have to test this more in depth when I get home.

2/7
Tonight on my graveyard shift I decided to activate the google api by getting a key. I updated the html files accordingly. I planned on doing some more testing with the RFFD but my work station was unable to display through the HDMI port for some reason. Instead I looked over google api documentation throughout the night getting myself

familiar with the libraries. After my nap I plan on implementing a full on search bar and test the api usage.
Later in the evening when I went to the gym I decided to take the RFFD for a test run to see if it will operate off of my mobile Hotspot while the server was on at home. I have a feeling it was rebooting itself several times since I noticed my phone saying a device was connected and reconnected quite frequently. When I got home to check on the most recent data, the heat map had no areas highlighted at all on my route and from last year; meaning the data must have corrupted on accident or something of the sort? I reloaded the old data points and the original heat map was back. Also side note: **the device must be connected to wifi or else it will keep rebooting constantly!**

2/10
After speaking with Guy earlier in the week, I asked him a couple of questions regarding how the RFFD was tested. Primarily, was it tested on an IPL vehicle in the past since the device is programmed to only operate when in a wifi network. Apparently it was tested from the original group members in their personal car through mobile hotspot NOT from an IPL truck. This will make it difficult to test on the trucks since they do not have wifi on them.
I looked through the code and tried to see if it's actually able to; it can if I remove all of the "sudo reboot" protocols within the main while loops. The scripts interface is a bit glitchy when I do this since the loop is supposed to break at system reboot, but is operational. Now the device will constantly be running until unplugged/powered down. This will be useful because gqrx and the gps module does not require internet in order to read the radio frequencies. Instead, it will operate as normal and create a log of the information in "packet storage" until it's able to connect to a friendly network and send the packets by bulk.
Guy also requested that I make a secondary device for one can be consistently tested by day to collect more data and the other for personal use to modify as I go. I partitioned another SD card with the exact build to install in a copy. Tomorrow it will be put on an IPL truck for the first time; let's hope the data read accurate, sensitive or insensitive frequencies or not.

2/11
Today I met guy before he went to work and he put my modified RFFD to an IPL truck today. The night before, I made it so that not matter what, it will not reboot while it's on during the day since I realized the device will work if and only if it's connected to wifi. It came to my attention that the device actually will not be hooked up to wifi at all times while it's attached the the truck due to security reasons. I also thought that the previous group had already had them on the trucks but I was misinformed; they were tested in

their own cars on their mobile hotspot as well. Wifi is supported on the trucks but since I do not work for IPL, it violates terms of service.
I went more in depth with Guy about how exactly is the device supposed to function and apparently I have to make it able to work offline at all times which is one of the reasons why it's a complete con right now.
During the day after work, I examined the code and created/planned out pseudocode to do a complete overhaul. My ideal propositions are: 1.) Make a while loop that will function as normal, however if any connection errors occur, I need to make it loop again but make it so it will not have any server functionality. Occasionally it should ping the server to see if it can upload data. 2.) I can make it completely function only offline and when it senses a wifi network in any form while successfully connecting to the home server, it will upload the packets at a certain capacity/continuously but able to continually run even after a disconnect.
After analyzing a code, I forgot one fatal error about the device in the field right now; once it's powered off and back on, it will freeze and not be able to continue onto the main loop since it's trying to connect the packet's in storage. I need to make the next modification to only send packets when it's within a network.
When I got back home, I put my plans to the test and rearranged the code to the best of my ability. Now I have to wait until tomorrow to receive the device and see what it found on it's single run and make it capable of running fully offline. Meanwhile during my work shift I added the google API key to my html and tested it; the "Development use Only" text all around the map is no longer there, so the api is in effect! I also reviewed what the usage is everytime I use maps. It's very abysmal right now and is costing me $0 dollars so I have a feeling it shouldn't be an issue.

2/12
Tonight between working and after work I looked over the html heatmap to see how the scripts work while the map is scanning for new data. The main function they run is the initializing map function and simply reading the data.json file with the lat, long points and place it on the map accordingly. The main bulk of the javascript comes from an api of the heatmap which is a starter kit. I could mod it accordingly if I choose to do so, like multiple points of the same.
I had some trouble getting the search box to properly function for a while, mainly because I had to change the original teams layout and add the extra scripts in the proper order so nothing breaks. Nonetheless, after finding the right areas and order, the search box works just fine but I need to find a suitable place to put it for the user. As of right now it's stuck behind the IPL banner! Tomorrow I'll be getting the RFFD device since I had an emergency to care of right after work. I also have my graveyard and will

be continuously running tests.

2/13
This morning I turned on the RFFD to see how many packets were stored from Monday; only 8 total pings were recorded… Mostly at circle center and that's it. I was wondering why this would be the case and then I realized that if the truck driver would get out of the vehicle and turn the car off, the device would become stuck. So the data that I got was a fluke, but now I have a proper understanding of why this is the case. Now this will make me rethink how I should restructure the driver file to only work offline completely and maybe at the end try to detect a wifi network. If not continue on.
I also spoke to my academic advisor about my progress on my project. So far he is still intrigued about the project and gave me advice, especially when it came to managing the data on the board so it can store backups before sending packets to the server, then the server side on building a database to secure the data instead of relying on one giant json file which the data currently is contained in. Obviously this is a very easy and insecure way for data corruption, so perhaps multiple data files should be able to be chosen on the client side to get access to a variety of heapmaps loaded in by the user. My advisor was also concerned of where the web server is hosted and highly recommended me uploading it to a secure site such as tesla since it can be publicly accessed instead of being accessed via only local network.

2/14 During my graveyard shift I focused on trying to get the device to work in an offline and online mode. I managed to figure it out after about 3 hours. The way that the device operates is similar, but with a slight twist. Instead of the checking past packet storage at the start of the program, it will enter a while loop first that will be active during the entire time it's powered on. There is an if and else statement that setups up offline and online. The if statement checks to see if the device is connected to a network by pinging the destination server. If it time's out, it will move onto gqrx and track the locations of amplitude spikes but instead of constantly updating it to the server, it will store it into the packet storage. This will proceed until the power turns off (which is a common occurrence in an operating vehicle.) On the else statement, this means that it is connected to the server, and the device will operate as before. If the device disconnects from the server it will reboot the system and see if it's connected to the network or not, if or ese. I tried to see what will happen in both instances if the packet is empty and not empty, and when you randomly disconnect from the network. The program ran just fine and rebooted when necessary. The only issues I had during the implementation is coming across a memory error probably opening and closing the packet storage at the wrong times. Near the end of the night I had an epiphany about how I could only record spikes when the vehicle is moving; I noticed how easy it was to implement the if

statements so I was thinking about implementing the signal strength > signal weight one with a && that checks to see if the speed is over 1 mph. The only issue is that I do not know if the speed is changing within the scope or within the scope of the function is called. That will be my objective this evening after my long nap.
Later in the evening I wasn't able to work on the project as much as I thought I would but my roommate helped me setup my new desktop that I will be using to run the server instead of using my laptop! I spend most of the night configuring it and am looking forward to testing it tomorrow morning.

2/16
Today I wanted to test out the && speed >=1 on if statement that checks for the spike. Essentially, if a spike is found it will only be recorded if the speed is above 1 mph. At first, I wasn't getting any spikes at all so I looked into the float speed variable. Right before the if statement I printed the exact value before it would go to the if. It stated that is was a -1 and would not change at all, so I knew that the speed would only be used when called through a function. So right before the if statement I tried to use the get_gpsdata function that retreives the long, lat and speed, filled the variables that I can used within the scope. This is when I found out the the speed can be used here within the while loop and in the scope. So, now that I had the speed working I tried the if statement again >= 1 speed, and it ended up working! Next I tried 5 mph since 1 was very sensitive after 15 minutes of testing the signals. After 5, the only time it spiked was when I was rigorously moving it back and forth quickly. So now, if a truck ends up at a stop light, it won't keep pinging the nearby area since it's not going above 5 mph. The range on it is about 100 feet in diameter, so once it starts to move it can easily pick up the signals that it could have retrieved when stopped.
|I couldn't get the device to work on my new desktop server side for some reason, so just like my laptop I will have to go into some configuration settings changing the firewall for either the router or the desktop itself. Once I figure that out, I can run the server 24/7. There are two things I need to look into a bit more on the hardware side before I can fully move onto the software portion of the project: 1.) Figure out a way to better manage data, as in have backups incase it gets corrupted 2.) Be able to automatically ping even if there was not a amplitude spike; this would have to be a timed process, so perhaps using get_time and using the system.time commands may be of use.
I also looked into ways I can make multiple RFFD's but I may have hit a bit of a snag. There are plenty of ways to clone a Raspberry Pi by the looks of it however I'm finding little to no information on cloning ASUS tinker boards. I came across a forum saying it's impossible due to the fact that it's against the Terms of Service for the brand, however there are ways to get around it. I personally think it's because I didn't properly boot the USB drive correctly as in having it on a bootable USB instead of just a USB that has the image. Still, this would be wise to get done for a quality of life change for both me and IPL wanting to make multiples of these.

2/17

Tonight I primarily was focusing on my preemptive planning for the server and what I should be looking ahead for in the near future. One of the main things my academic advisor Dr. Zheng highly recommended me doing is putting the web server on tesla.cs.iupui; the only issue is that the web server is currently set up to run on node.js locally, which is as far as I can tell embedded into the html/javascript portion of the IPL heatmap. Also, the server.exe that I have to run in order for port 9930 to communicate with the RFFD is at fault as well, because server.c is in the same directory as the IPL heatmap node server; they have to be in the same directory in order to communicate, while server.c is running in the background. So I cleaned up some of the HTML portion to make the search bar be able to be implemented in a clean fashion and started to look into how the scripts run again and where it obtains the information for the data.json file.

To do something simple, I uploaded the server software onto tesla.cs.iupui and changed the html file to index.html in order for it to be a public domain. However, for some reason it's unable to detect the data.json file even though it is in the same directory with the data.

So I have a few option here to think about; should I find a way to adapt the old server communication or try to overhaul it entirely? When the RFFD sends it's packets, it's sending it to an IP address along with a port number, 9930, and awaits for the program server.c to obtain the information, do a "handshake" and notify both parties it was safely delivered. I have to remove this or upgrade it and then find out a way for it to ping my school's web server cs.iupui.edu/~josvaugh/RFFD for packet transfers. My professor mentioned that he himself has done projects that runs C/C++ code in the background of his websites that runs applications, so maybe I should seek out more advice from him about the subject.

2/18

Today I used remote desktop to start downloading some software that is maybe in the right direction for building a database within the tesla server. I ended up spending a good amount of time studying up on php with it's syntax and how to integrate databases within an html doc. Php works by the looks of it on the tesla server with a simple hello world program, next I want to create a test database using MySql which was one of the installations I had downloaded. I definitely need to read up on more documentation for it since it has a workbench but I need the database to be on the webserver and or my desktop to store the data. This is to essentially gut server.c entirely and overhaul the data storage system. I will also need to find a way to extract the data and use it on the heatmap. Meanwhile, the device is currently on an IPL trucking working offline; I'm worried about how the data storage is working with that since the max storage it can handle is 100,000 bits/bytes, which is the equivalent to 6,250 pings but since it will only ping when it is over 5 mph, that will decrease the workload for sure.

2/19-21

The past few days have been a bit slow due to other coursework and a shift in my job schedule, however I've been doing research on PHP scripts and reviewing SQL since it has been about two years since I've had to write queries. I understand the idea of how to create a working database within the heatmap site however I'm unsure on how to send the packets to a database perse since my website will be hosted on the tesla server which is on a public_html domain instead of a website provider where you simply have a "add database" button that doesn't exist in public_html. Meanwhile I spoke to Guy about getting the RFFD back on Thursday and his truck driver just plugged it in and turned it on today; I'll be waiting to get it back Monday to see how the field tests went. As stated in my previous post, I plan on gutting and overhauling the server-side portion of the website. Since I have not done this myself in any class before this (besides writing queries in Microsoft SQL with a database provided for me), I have not had to build one from scratch. My professor has a running website using the Tesla domain so I do plan on meeting with him next week after I made more test databases and php scripts over the weekend.

2/25

Over the weekend I had issues with another project that consumed most of my time unfortunately. I ended up getting the device that was out for field tests from last week, about two days worth of data, 2,000 lines of code in the json file with the plot points. I looked at the data, loaded it into the map and was trying to display my results. I actually did not see any form of heatmaps except for the initial two from the previous data set (a place for me to reference). Comparing the data, the points should hypothetically be appearing on the map, so I looked into the original data file and honestly didn't see that much of a difference. I even ctrl + f the exact strings that should be there, such as lat, long, wt, epoch and they all had the same occurences. So this either means that the data is inefficient, there's a break in the code or something external happened to the recording of the data. Guy told me that the driver requests a strong antenna since the previous one broke it's magnetic part. Perhaps this tainted certain parts of the data causing it to appear blank on the map.

2/26

After writing a couple of php simple scripts and familiarizing myself with the MySQL workbench software such as watching videos on youtube, I've been compiling a list of questions to ask my professor tomorrow since I have my weekly meeting with him. One of my main questions is how he had his website gain access to his database while on tesla, how exactly is php is supposed to be used in this context, and whether or not I should publish the website on tesla since once I graduate the site and setup would be rendering useless for future iterations.

Since I am entering uncharted territory with creating a server and database from scratch I want to make sure I do it right the first time.

2/27

I met with my capstone advisor today and started to ask the questions I compiled. My professor realized that I have a very limited amount of options in tesla particularly with permissions. Say I wanted to update my database on the tesla side; he would have to approve every commit I would make to the database. So he recommends that I run the server locally, using the current setup I have such as node.js, and make sure the site can run just fine locally and implement the database on the same local machine. In a sense it's similar to have the setup is now except instead of storing all of the data in the json file, I would add it to a database.

He made a couple of points to make for me clearing my questions and concerns, which is essentially what I learned from my meeting:

- Store the database on my local machine (for now).
- Php scripts are meant for the client side; say the user wants to use a certain set of data such as date, time, strength etc. This would also be where the admin team can edit the data as well.
- Find a way for my Server.c executable to write to the database instead of the json file OR have a program that reads the json file INTO the database
- Make the server hosted on my local machine for now and ponder for third party support such as Google Cloud and Amazon Web Services
- Implement EVERYTHING locally then upgrade to a better host once I figure out the scripting
- The device sends UDP packets; he recommends TCP since it's a more secure connection

This meeting definitely did help since I brainstormed my ideas; originally it sounded a lot harder than it actually was but really I just have to upgrade the current specs even more.

2/28

Tonight I had to get ready to pack for my move this weekend so I didn't have much time to work on the device tonight. Instead I invested some time in creating a AWS free tier account and view my options I can have for a live web server or database. I looked into Wordpress, Symfony, and Lightsail to get a sense of what the backbone software of the server should be. I think I'm leaning with Lightsail with a combo with Symfony since that is mostly what supports php client side programming. I noticed that there is a lot of resources I can utilize with AWS such as free tutorials, classes and guides to setup a webserver for free.

3/04

I didn't get to work on the RFFD this weekend at all due to me moving and unpacking, but I did do a lot of planning about what I plan on doing in the future. Tonight I wanted to resolve my most recent data find with 2700 lines worth of the data.json file. I thought it was a fluke since I

didn't see any pings on the map however I decided to turn the Map mode on instead of Satellite to get a less detailed view since I was trying to look for vibrant colors. I noticed some small blue circles going throughout 465 that actually created a trail all the way to Carmel, somewhere I was not expecting the truck to go. Based on my data, the 5 mph adjustment definitely reduced congestion on the map! I only spotted 4 clusters of red zones on the map, but they were tiny in comparison to the original data set which was basically red at every stop light and intersection.
Guy came in contact with me today and asked if I could add a feature that which be helpful to implement which is in addition to the device being able to work online/offline and scan only when it's moving; he would also like it to ping every two minutes regardless if it catches a signal. This may prove to be a challenge to me for just a couple of reasons, 1.) is that the way the board is setup now is that it **ONLY** records signals that hit a certain threshold. This might be difficult because 2.) I would have to figure out how the program senses it has been two minutes inorder to record. I have a feeling the answers is in the time.h library but since I've only worked with it once in the past, I may be a bit familiar with it.
Since this was one extra thing that I had to implement and I'm trying to figure out where to start, I created a draft set of UML diagrams that can help with planning in the near future. I made 4 stage:

- Stage 0: The initial given code and device
- Stage 1: What I currently have implemented
- Stage 2: What I plan on implementing
- Stage 3: The final end goal if everything goes smoothly in stage 2 and between

I showed Guy the diagrams to visualize where exactly I am at, especially with the new plans that I will be attempting to implement this week and Spring break. I expressed my concern with the AWS service since it can get pricey for certain servers if I reach a threshold of traffic/use on my website (which is LAMP php based.) The database is way too expensive, the cheapest option is $700 monthly. However the LAMP site has a free threshold; if I go over it, it can cost anywhere between 10-70 dollar\s monthly depending on what my needs are (similar to the Google API.)
He went over how cheap IUPUI's IPL projects are compared to their actual development team, so I have the option to launch an AWS based site next week if I choose to do so!

3/05
Tonight I started to look more into how MySQL Workbench works, so I sat down and watched a couple of tutorial guides in how to create a simple table. In this table is the same data that I am using from the data.json file. Most of them are floats and the TruckID's are varchar(20)'s.
The way MySQL workbest works is quite different from the previous times I've worked with databases. Everything that I need is right here in the application; I have access to the queries, the visual representation's of the database, and even the server admin portion that has all of the logs and activity from using the database.

In the past I used Microsoft SQL and erWin modeling, 2 separate softwares. In MySQL, everything is in one software. Within MySQL I can choose where to connect the database to, which is when the PHP portion of this project will come in. The php portion should be apart of the website, but obviously link it separately and not within the .html document since it can be viewed publicly.

3/6

Today I decided to try and connect to my testing RFFD database by adding php scripts into the heatmap html. I believe something went wrong with initializing the database. I might be missing a couple of connections to the server that could make the database live. I should add sample data to the database and connect to it within the SQL queries in Workbench. Once I verify that it's working correctly, I can then continue working on it by connecting it the html documents.

3/7

Today I woke up to find that the entire upstairs of my house had no power, so I could not work on the database on my home desktop. Instead of looked into tutorials on how to set up a SQL database through youtube. It definitely is not live so when I try to receive the connection to the database, it's actually unable to receive anything. I have to first initialize a connection to the server to see if it's up and running, then connect to you via php scripts in html.

3/11

Since Guy me I can go crazy with AWS I've been looking into implementation guides to start a PHP website using LAMP. A majority of my time was reading through and practicing the examples they were providing. One of them was simply how to launch an application using Beanstalk which is essentially a website generated by an engine of your choosing, such as node.js, php, and Ruby, etc. I learned how to properly launch, update, and even terminate an application if I so desire and how to modify the website. I also followed the guide and got to upload and connect a relational database to the php site. Obviously it was empty, and hypothetically it should be free to use monthly depending on how much data I store in it, how much I access it, and how many users use it as well. If I want it to have 100% uptime, security and all of that, it would require a lot more money just to run it per hour. The guide I was reading says that it should take up to 2-3 hours to "complete it" and by that, it mean's read through and implement all of the provided examples. I worked on it for about an hour and a half but most was getting the general idea of how the console center works. I will definitely have to refer to this guide as I build the site.

3/14

After learning how to properly create a php based web page and connecting a database, I decided to try and see what would happen if I took my heatmap based site and imported it inside of the

html portion of the php document. Unfortunately, it did not end up going through correctly when I tried to update it. I need to to try and make a basic website first, so start small, and implement a proper user interface instead of having a map blow up in the users face when they access the site. It should have a home page, map page, data page and admin page. There should be a login feature but we will worry about that later.
I was a bit confused with how the apps are set up in the console page so I looked more into what other options I have, such as wordpress, the static version, and a simple site. The differences in all of them depend on what the designer has in mind for the users. A majority does not support php and server side programming so most are totally useless. This will encourage me to really overhaul the Server.c program that I have to run to transport the data to the html site like before. I need to find a way to use these applications that I've been learning about and have them work together with an unknown device such as the RFFD device.

3/18
Today I looked into setting up most of my current setup to run locally. I asked my coworker(s) what there opinion on AWS was about making a php application and were actually surprised that I am using it this late in the semester considering that I am relatively new to server-side programming, php and making databases from scratch alongside unknown hardware I have not used. Perhaps it's a bit too naive to start using AWS right away, maybe for providing a live server for the website only? The only issues I'm having is having it run just fine using the php engine since I originally wanted it to be on the Tesla server (which supports it), but I still have no idea how to "link" other web pages together, primarily html since I can't even get a basic html website running via Beanstalk. Maybe I just need to update myself with the documentation, but time is of the essence and I have to decide, so for now on I plan to use my local system for 1.) creating and hosting the database 2.) hosting the web server 3.) have the application load data into the database using server.c.
While I was trying to figure out how to run php scripts under Node.js' server domain I found out that it does not support php at all, unless by using specific libraries that I myself am not familiar with. So currently I need to find an alternative to node.js. There seems to be a road block along every step of the way. Tomorrow I will have drop the RFFD with Guy so they can do more field testing with it tomorrow. Essentially I'm trying to get more data to use and see how accurate it is while moving and not moving.

3/19
Didn't get to work on much today, but I did give the RFFD to Guy for it can collect data throughout the week. I also brainstormed with one of my peers after work to get a sense of where I can go from here since I found out Node.js isn't compatible with php, which is how the server is currently live locally, so I need to find another way to have the site be local while running php. I spent most of my time thinking about that and googling different methods. I've also been

reading guides on how to get php to properly have it run on my computer. Perhaps I need an ide to run it? The only way I can run it by using it on the schools tesla server which isn't going to be where the database and site will be hosted.

3/20

My academic advisor was out of the country still in China and I had hoped he would have been back by now so I was not able to meet him to discuss options about launching a local server that's capable of hosting php applications.

I attended my other class and happened to find a friend that had taken a server-side programming class and asked for his advice on how he would make a php based website on my machine. He recommended using apache and possibly oracle frameworks. The last time I used apache was 2 years ago but it was in Linux so I did some research on how to run it through windows. This lead me to look into XAMPP, which is I guess a way to make your computer a local host similar to node.js, but capable of connecting php, mysql, and an assortment of other web applications together.

I had to do this all remotely, so I was a bit limited on my capabilities but I ended up getting a simple hello world html site to work! I then copy and pasted the heat map into the document to see if it would compile and it did! The only issue is that it runs off of the local host Ip address, 127.0.0.7, now I have to see if there is an alternative that uses the IPV6 public way to be able to access it from outside the local host like node.js.

Now I know how to set up apache, so now I can start running test php scripts and eventually mysql database queries.

3/22

Today I did not get much actual work done since I was preparing for an interview with a nearby company, however during the interview I had an in depth discussion about my Senior design project, primarily about how I plan on setting up my web server since node.js did not work and I'm having difficulties with AWS LAMP. The senior director highly recommended staying away from AWS unless I plan on having a lot of traffic flow on the server. Since it's ideal to have maybe 10-20 users at a time accessing the site, AWS is a waste of time, energy and resources. Instead he insisted on trying out supporters such as Digital Ocean and Site Ground which basically will charge you monthly for full fledged server and database support for small traffic servers. In fact he said it would be as simple as using my local host setup and a "drag and drop" file system for my website. This weekend I plan on investigating further on his recommendations since I am at a loss with a public web domain with AWS.

3/23

Saturday night I decided to start making an exoskeleton of the main site. It's fairly basic right now and I'm still trying to figure out how the XAMPP domain works. It's similar to the

public_html like Tesla. It does support php since that was the first script I ran and I tried to set up a basic database but I was not successful in connecting it via php. For some reason I don't have a password in order to connect to it even though I made the password fairly simple and added it. For the initial site, I referred to one of my older sites I created. I linked up the heatmap so you can access it from the home menu which is will be an introduction to how the project works.

3/26

This week has been and will be pretty busy since I have a couple of interviews lined up and am currently starting my final project in Multimedia. While I was at work today, I had some downtime to start looking into the errors I was having with the mySQL connections. It looks as if I should flush out all privileges to my account show I can start again with it. Later that evening I tried to resolve the issue but didn't go far. Instead, I watched the first 2 hours of a "course" over mySQL from start to finish on how to implement a relational database through the workbench while also going over old concepts that I have been needing to review. I didn't get to finish it completely, so I will finish it tomorrow while also meeting with my capstone professor to discuss further implementation of the php side and mySQL portion.

3/27

I met with my academic advisor and updated his about how my project is going. Overall, I just need to get the database to connect to the php portion of the html site and I should be fine for one of the final parts of implementation for this project. He provided samples of how his students in his research topic used php with videos, demonstrated a good idea of how my site should provide the data for the user. He also told me that if I do have any issues with the database portion of the project to contact Ligma as soon as possible, email or in her office.

Shortly after I finished watching the 4 hour course video. Once it started on the actual implementation part with the server and relational databases, I decided to uninstall my current mySQL workbench to start fresh. It's safe to say that I got it to work and connect to the sample database! Now that it's connected, I need to test further with a test php example in apache! I also was sure to write down all valuable information for further connections, such as the connection, username and root password incase I need it.

3/29

Tonight I had a bit of time to work on the database portion of my project. I had a lot of difficulties simple using the query command like so I decided to start small and make another database where I could practice creating tables from scratch, messing with the different data types, and inserting data one row at a time. I also had to end up googling how to use some of the different functions within MySQL workbench like making what's my default database to use. It was a nice refresher to experiment with another IDE instead of visual studios. I left off with

experimenting with making the RFFD database, but I had to brainstorm how I could approach it since we only have one truck and have to determine the primary key.

4/1

Today I primarily focused on getting a fully running database based off the RFFD. I had to make a primary key up on the spot so for now (until we get more trucks), each ping on the heatmap will have their own unique key. Once I get an idea of how many other main columns like a TruckID, I will adjust accordingly. My main focus was finding a way to add the json data to the database instead of obviously creating 500+ lines of code at once. I managed to find an online "SQL Generator" that completely dismantled my json file and made it into a sql query. It's a nice band-aid fix until later. In fact, I looked into what type of functions php has for making said queries. So if I wanted to hypothetically add the json file at any time, I could upload it to the website and it will store the data in the database! Now I just need to figure out if there is a way that the heatmap.js can view the points from the database server instead of the data.json file… Anyways, I managed to upload about 300 rows into the database from the March 11th data set and even tinkered with the different weight's it picked up.

Overall, I learned a lot about SQL today and Friday. My only issue that I encountered was keeping MySQL online in XAMPP which I ended up trying to research at the end of the day. It may be a bug with the cache, however I will start tomorrow with connecting it to the test php application I have through apache and xampp. Once I get it running properly, I will hook it up to the main index of the base site.

4/3

I spoke with my advisor about the progress of my project; I told him that I have an up and running database that I been experimenting with for the past couple of days, mostly running test queries and finding out alternative ways to upload 300+ lines of data all at once instead of one by one. He wanted to know what was on list next in my progression which was to use a web server host to host a live website to continue the testing where I can fully utilize software that can use php. For some reason I keep having extreme difficulty with my local machine connecting to php so I discussed with him that I plan on using siteground. Shortly after talking to him, I spent a good amount of time looking up youtube review's and guide's on how siteground works and how to set up databases and use php. Luckily they offer tutorials on Wordpress, MySQL, and connecting php to MySQL databases all within the same network. Essentially all I have to do is create the database within the server's network similar to how I use MySQL workbench but it's all in one place instead of worrying about it on my local desktop through XAMPP. The price for hosting a website for a month, trial period is only $3 but I have to buy rights for a web domain so all in all it was about $30 bucks. If I'm not satisfied in the next couple of days, I can always request a refund and find an alternative if need be. There are many tutorials on youtube and within the website so I have a good feeling about this. One of the requirements for IPL is to have a live website up and running by May that they can pick up and use at their discretion. Now it's

time to make the website go live and work on php applications and setting a database up on their server instead of my local machine.

4/4

www.ipl-rrd.com; that is the domain! Overall I paid $35 for the site, it's $2.99 a month however I had to buy the rights for the web domain, but they offer a money back guarantee so I will be sure to use that shortly after my project is finished. I went for a Wordpress template that is dedicated to a business oriented website; I have little knowledge about Wordpress, but I made some slight adjustments to identify it as the RFFD site. The overall design of the backend is quite interesting. I had to use the phpMyAdmin tool in order to see and view what's inside of the database currently. I made a mock database that was very similar to my local host setup, now I'm trying to figure out how to implement php within the main site. I believe the best way to view php is by comparing it to how javascript operates; make buttons such as "Connect to Database" to initialize the connection for the users. Then, depending on what they want and or need, to make it so they can filter out the data instead of manually looking at it. Essentially, the way siteground works is that you have your front end HTML site which is powered by wordpress. Then on your back end, you can pick and choose what you font, however, the wordpress I'm using is actually php based, so all of my commands should work within the confines of the site!

4/8

After work I decided to review portions of the cPanel and Dashboard (which is the admin parts of siteground) and looking over the database that I created, taking note of the username, password and connection names. My biggest qualm so far with siteground is that the website theme I chose is very basic and driven mostly by the Wordpress engine. There isn't many tutorials on how to add PHP to it on sitegrounds site so I found out that you have to install plugins in order to add PHP lines of code. I was also trying to add multiple pages such as the Severity, Map, and Admin page as templates but the way you setup a new page in Wordpress isn't as easy as I thought it would be. For the rest of my evening I looked into tutorials in how to add php snippets to my existing website, but yet again I'm still confused on php in general. I found a website that teaches basic fundamentals of php that should be apart of every site like email, accessing a database, displaying data tables from a database, etc.

4/9

This evening I was curious to see if there is any way to customize my website to make it so it's not generated and engineered by Wordpress. So to start off, I did some research to see how I can access the files on the backend (even the original default wordpress php applications) and I found out that sitegrounds admin panel actually has a file manager similar to Tesla where all I have to do is specify what my index.php(html) is and it will make that my home page. I tested this theory by renaming the original index to something else and uploaded my exoskeleton website that I was previously using from the locally hosted portion on my desktop. Just like that,

it worked! So now, I can drag and drop my files that I have been testing locally on a server side instead of local. After that I tried to find out if there is anything in particular I have to do to connect php and mysql together but still, they use the “localhost”, “username”, “password” setup that every tutorial goes over. I should search more about this in the admin panel later tomorrow.

4/10

Today I had my weekly meeting with my academic advisor about the current progress of my project. Essentially I explained to him that I spent the past few days figuring out how to work siteground properly, such as how to use the admin panel and the dashboard through the wordpress engine. I also showed him that they also have a public_html domain similar to the school tesla server and told me that this should in theory work just fine for my project in general. I told him my primary issue was figuring out in theory how php should work which was very similar to my realization the other night. Javascript will basically configure the UI on the frontend, but php is the server side portion that can use javascript UI to connect to the server side. Say you have a button that says “Connect to Database”. You would have a javascript button that visualized the button and then perform a php script that connects to the database on the server side. He ended up giving me two examples of his research code from his grad students to give a solid example of php scripts in action which is what I have been struggling to comprehend. During my work shift, I looked at his code and analyzed it to the best of my ability.

4/12-13

Over the weekend I tried to test the waters by connecting test php scripts to my database through the file manager in the admin panel of siteground. I managed to finally connect to the database just fine after rigorous trial and error of finding the exact name of the server and database. I spent a couple of hours figuring out the SQL code to display the data in tables. I at least know how to connect it just fine, so I can start implementing more php sample code getting a sense of how I can use it to my advantage. On Saturday, my desktop computer had some trouble with the sound card which lead me to having to reinstall and update my computer. For some reason, my computer would not turn on afterwards, so I had to move to my laptop instead. I can’t access my files for the time being but it’s not the end of the world since the important ones are now on the server side of the site instead of how it was originally, being locally based. It’s inconvenient but I can manage. So now I just need to run php code to do what I want it to do for IPL’s wants and needs.

4/15

Today I ended up hanging out in the library and really wanted to crack down the issue I was having with php connecting and displaying data from the database. I did end up finding out out to display data from a database to any and or all row’s if I choose to do so. The issue I noticed was the fact that I was using mysql, NOT mysqli, which is the improved version for php 5 and

onwards, which is was siteground supports. So after figuring out the minor details such as that, I tested out the various amounts of data manipulation tricks. I've had to also learn a couple of ways to utilize php, similar to how I was finally figuring out that php is essentially javascript programming but for the server side part, I also found out that you can have your scripts separate from the html/javascript file and call it within the same directory. I assumed most php applications would have to be on the same file, the more you know! I also messed with the different $sql arguments I could use when you press a certain button. Say you want all weights that are only 50 and up, you can do that and it'll retrieve all of the data points that meet those conditions. I would like to make it customizable, so I have to be careful for sql injection is not possible for the security portion. I also made some adjustments to the main index.html to make it to more presentable to how I would like it to stand out. Currently there's a Home, Heapmap, Serverity and Admin button. One of the last things that I started to research was how to have the heatmap.js read from the database instead of the json file. Meanwhile I have to find a way to upload or change the way the json file is read so it's capable of entering the database.

4/16

Today I kind of opened a can of worms and wanted to see if I could get the Heatmap file to work by instead of using the json file it was originally programmed to do, to instead read it from the local database in the admin panel. It was fairly difficult to find good documentation on it, so I had to thoroughly comb the web for scripts to run it. Essentially you want to obtain the data by using an SQL php script and then initialize the points based on the data you retrieved. I could not end up getting this script to work, but I have a broad idea where I should go with in. In general, the front end portion of the website is about 60% complete; Once I get a running heatmap, an ideal severity list, and and pseudo usable admin panel, it should be complete. The backend could use some improvements, but I need to get it running in order for my project to be "complete" in my advisors eyes.

4/17

Today has been rather busy for me since I had 3 interviews and working on another final project (while also working) so I've had trouble finding time to work on this project in between. I wanted to go into more research finding out how to display the heatmap values so I started to look into github files and libraries to see if I can find one to assist me in this issue. I'm still stuck on getting the points to work, but I do have it so the script does "run" while not crashing the webpage which was one of the issues last time. Apparently it's very sensitive with catch, try syntax. I spent some time trying to fix it but was not able to resolve it.

4/18

At first, I tried to find a way to display the data with the heatmap but I kind of realized I have a huge underlying point I'm not really looking into which is how the data can actually enter the

backend portion of the website. Somewhere within the Admin panel, there needs to be an upload file option where you can add files, such as the json data to the database. I did find a few youtube videos that highlighted how to do this using PHP such as how to add and Upload File button and where it would go directory wise within your public_html. My goal is to make it so that almost all of the front end portions needs to use the database as the source of information, not the json file. So I spent the majority of my time trying to find more PHP tutorials and started to develop test scripts that won't affect my files.

4/19

Today I didn't get to spend too much time on the RFFD, due to a priority in another final project, but I did end up migrating most of the IPL files over to the server so I do have data to tinker with. I noticed the heatmap only works in some cases, primarily glitches with images and the map being displayed (which has only happened after I tried importing the database portion.) In response, I added the image sources and the supporting heatmap.js libraries that was originally included which might be affecting the overall displaying of the data. I also discussed with Guy what is the current status of the project; Overall, I would say about 50-60% of it is complete, in the next upcoming two weeks, I definitely feel like I have the time to make a presentable Prototype of a website for them. I also told him that it has been getting a lot of attraction from my interviews with companies and asked if I could still work on it as a side project during the summer to keep building onto it since I will be graduated by then. He said by all means, you may since this may or may not be the last iteration for them on this project; I have been a handy tool detecting whether or not my research is enough to invest into it more.

4/22

Today I wanted to look more into how to make a filter that was similar to the original heat map but to filter the internal data the user can view from the database. I searched a variety of videos using php and Javascript to get an idea of how I can approach this. I also watched a lengthy video on how to make a login screen for the Admin page for they can login that. My ideal thing to do for that page is to maybe import phpmyadmin and an upload file to insert directly into the database, which I found out a broad way to do it on another introductory video. Overall, I spent most of my time trying to learn these methods from different youtubers to apply in my own ways.

4/23

Today I had a couple of interviews I had to attend and a group project to work on, so I did not get to spend much time today on the server portion of the website. Instead I cleaned up more of the front end and tried to implement a data filter on the Severity tab that's similar to the heapmap tab. I couldn't get it running just yet but I would like to say I'm going in the right direction. I hope to implent a top 10 wt list, most recent list, and perhaps one more.

4/24

I met with my academic advisor today to discuss how the project is coming along, and showed what I currently have. He was a bit underwhelmed and told me that the original one I had earlier was better for presenting. Since the deadline is next week, he wants me to focus more on a practical demo (even if it's not correct on the backend) just to show and present to the class how the end result should (and hopefully) look like. As long as I have a detailed Home page, a working heatmap, a list of any sort, and maybe just a simple login screen (with nothing in it,) would do just fine as a submission in itself. We also discussed at length of how difficult this has been for me doing it solo since it is an entire Frontend, Backend, and Smart Device I had to work with over a 4 month span of time where I had to also learn Server-side programming by myself. I did also let him know that I am gaining a lot of attraction with this on my resume which is impressing employers and also I plan on working on this on the side after graduation if I don't land a job by then to boost my resume even more! So my plan for now is make simple runnable php scripts that will display presentable information in regards to the requirements that was set out for me at the start of the semester.

4/29

Tonight I worked on the severity list being presentable for my presentation on Friday. It properly shows every point that is within the database. Future implementation options is making a smarter filter for the data. Since I'm having a lot of difficulty with the heatmap reading points from the database, I will stick to the original setup and use the json to present the data in a clear manner. For some reason the google maps portion kept breaking so I need to evaluate the situation and check on it tomorrow. I also added all of the source files from previous attempts that work into the same directory on the server.

4/30-5/2

During this period, I went into high gears mostly making the front end look great! I decided to implement a nice header I used from the previous home page and used it on all pages: Home, Heatmap, Severity and Admin. I also modified the CSS page for it can be used on almost all of the pages as well to go along with the color scheming. During this time, I also noticed that my search box was not actually working properly so I went EXTENSIVELY trying to figure out why it's not behaving correctly with my heatmap. After about 4 hours of trying to fix it, I decided to forget about it and worry about making the placeholder for the Admin panel. It's pretty simple as well, currently it is setup so it only accepts root and password, it'll notify if it was successful or not. In the meantime I started to pull my presentation together and presented on the 3rd of May.