So far:

* Read the intro paper
* Skimmed waterfall.py code and traced the API back to its install location and read the documentation there
* Understood the api and values as well as the FFTs and its output and API by a bit of experimentation and more code reading of waterfall.py
* Tried using winsound for audio output of sonified radio waves (range given in lots of documentations and waterfall.py) but python version to new/incompatible so moved to pyaudio which had a PEP 517 error which was fixed by quick googling and doing “pip install –upgrade pip setuptools wheel” before installing pyaudio as sudo, re had issue so used below 2 links total ish
* [python - Could not build wheels for \_ which use PEP 517 and cannot be installed directly - Easy Solution - Stack Overflow](https://stackoverflow.com/questions/64038673/could-not-build-wheels-for-which-use-pep-517-and-cannot-be-installed-directly)
* [python - ERROR: Could not build wheels for pyaudio, which is required to install pyproject.toml-based projects - Stack Overflow](https://stackoverflow.com/questions/73268630/error-could-not-build-wheels-for-pyaudio-which-is-required-to-install-pyprojec)
* Tried to get single tones output using pyaudio since winsound is outdated on python3, heres the link:
* [audio - Python playTone(freq, duration) command for Learning and Teaching - Stack Overflow](https://stackoverflow.com/questions/40810710/python-playtonefreq-duration-command-for-learning-and-teaching#:~:text=import%20pyaudio%20import%20math%20def%20playTone%20%28freq%2C%20length%29%3A,play%20frequency%20for%20given%20duration%20playTone%20%28frequency%2C%20duration%29)
* Found out after getting pyaudio working that it has some issues with the sound card not properly being detected, but found the below for pygame version which does work:
* [Python simple audio tone generator - Stack Overflow](https://stackoverflow.com/questions/56592522/python-simple-audio-tone-generator)
* Found how to set volume here (using prior link figured I was using a ‘sound’ so I used that code):
* [python - How do I change the volume of the sound or music in PyGame? - Stack Overflow](https://stackoverflow.com/questions/65247656/how-do-i-change-the-volume-of-the-sound-or-music-in-pygame)
* Did a few things, needed some python language specific checks. Found here:
* <https://pynative.com/python-check-user-input-is-number-or-string/#:~:text=To%20check%20if%20the%20input%20string%20is%20an%20integer%20number,using%20the%20int()%20constructor.&text=To%20check%20if%20the%20input%20is%20a%20float%20number%2C%20convert,using%20the%20float()%20constructor>
* After the work above had a simple sonification program set up and working with a mediocre terminal interface. Was sent on work trip to ATCA (Australia telescope compact array) and learnt radio interfremetry as well as a lot of radio details from documentation and lerant AM FM and basic intro to radio interfremetory from resources provided by mentor to finish off the next 2.5 weeks of studentship.
* On 4th week start had todo admin safety tutorials and continued work on how to combine the audio clips from radio back into a single slightly freq separated signal as well as testing if audio output works on subwoofers/vibration pucks for tactile feedback as well as the functional auditory feedback
* Got multi synthesis working using these links:
  + [python - Pygame mixer only plays one sound at a time - Stack Overflow](https://stackoverflow.com/questions/15385727/pygame-mixer-only-plays-one-sound-at-a-time)
  + [pygame find channel returning nonetype - Search (bing.com)](https://www.bing.com/search?pglt=41&q=pygame+find+channel+returning+nonetype&cvid=44bcb254ed814e97b35404640cc5c4d5&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIGCAEQABhAMgYIAhAAGEDSAQg4NDI0ajBqMagCALACAA&FORM=ANNTA1&PC=ACTS&ntref=1)
  + [pygame set amount of mixer channels for synthesis - Search (bing.com)](https://www.bing.com/search?q=pygame+set+amount+of+mixer+channels+for+synthesis&cvid=795b1dd94f004850a9a0125137bbcb51&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIGCAEQABhAMgYIAhAAGEAyBggDEAAYQDIGCAQQABhAMgYIBRAAGEAyBggGEAAYQDIGCAcQABhAMgYICBAAGEDSAQg2NzAxajBqNKgCALACAA&FORM=ANAB01&PC=ACTS)
* After it worked discovered a case where the sound waves made from the frequencies weren’t playing till stop length so googled the problem and tried this:
  + [python - How to run sound loops in sequence with pygame? - Stack Overflow](https://stackoverflow.com/questions/61067003/how-to-run-sound-loops-in-sequence-with-pygame)
  + The above link provided how to get the sounds to play looped then I just added a pygame.quit() command after the pause to allow the sounds to stop playing on exactly the user input duration since otherwise theyd play for in-exact amounts
* Next looked into maybe changing the UI to allow more user control over how to play the sounds and the NFFT variable to control how many channels were made by the FFT, further looked into how to asynchronous the functions and the control flow for those to ensure most optimal usage of computing resources
* First had some struggles with actual run time being horrid so did a search for slowest parts using the timing method from below:
  + <https://www.programiz.com/python-programming/examples/elapsed-time#:~:text=In%20order%20to%20calculate%20the,which%20gives%20the%20execution%20time>
* After identifying the issues above noticed could reduce sound synthesis by factor of n from quadratic to linear time using geometric series summation on a series of sin functions and eulers number
* Python on windows is a struggle, but found these commands for both python and git/bitbucket
  + “python venv .venv” to create a venv with the file name .venv
  + “.venv\Scripts\activate” to activate the venv but might have tweaked path name depending so open the venv file created in previous command to find exact path
  + “deactivate” to turn off the venv in python when youre done working
  + bitbucket is identical to GitHub in CLI usage and commands, might have to do “git remote add <remoteName> <remoteLink>” and if you accidentally upload the wrong branch to the wrong remote branch name when doing “git push -u <remoteName> <localBranchName>:<remoteBranchName>” which can be done without the specified remote branch name then simply add an “-f” to the end of the command to force the local branch copy to over-write the remote branch name
* got python working had to restart but otherwise made the program faster by generating sound arr for 1 period rather than a whole second causing a speed up by a factor of frequency of the sound wave, adding asynch now including to the pre-made functions for pfd and fft and maybe data fetch since I can and need it
* so used this link: <https://stackoverflow.com/questions/50757497/simplest-async-await-example-possible-in-python> to create a basic fire and forget asynch loop but that wont work for my application since I need to sonifi immediately once the previous audio clip ends so I might use a queue instead like here: <https://docs.python.org/3/library/asyncio-queue.html>. Which would also involve making the asynch await for the previous sonifi to finish be in the next call to that function rather than the parent function which would also have its own asynch processing but to be figured out for now. Might actually be simpler to have 2 functions 1 to create sound array and another to fire and await them only which I like more as an idea
* async functionality working and heuristically dialled in parameters that effect program speed to match and be functional to supervisor standards, moving onto making the presentation slides