1. Agreed to create web based Password visualizer as my research project.
2. Agreed on using JavaScript/html/css mainly but can also use any backend if I am required
3. Research what does “Edited Distance” mean
4. Create a canvas with a centre of the password.
5. Read supervisor paper, think how to do it myself, try possibly doing it myself, then I will get the code but i will try to still create mine. Try to generalise it.
6. Think of an efficient way of displaying passwords.
7. JavaScript code. (look at visualisation libraries for JavaScript)
8. Test out some libraries from NodeJS possibly

# Paper PSV Notes

1. Passwords are often targeted to access security systems and even though people have made prevention techniques to cracking them (salting and hashing) but still the weak link is the human who chooses simple passwords making them easy to crack.
2. PSV (Password security visualizer)
3. PPC (Proactive Password Checker) on the client side, they are tools which indicates the strength of the password to the user whilst he is creating it.
4. RPC (Reactive password checker) on the server side, tools which often scan the system database and sending cracking attempts to search for weak passwords, information about the passwords is then send to administrator or/and affected users.
5. PSM (underlying algorithm to detect password strength) password meter.
6. There are studies shown that UI of PPC does help users with creating stronger passwords
7. Some people are confused about the PPC and don’t know why its there
8. There are people who know how to make stronger passwords but don’t.
9. Telapath words was introduced, even though good way to get people to use stronger passwords, they found it hard and annoying to use. Prove that again its hard to make a good PPC.
10. (PIUs) (Password Information Units)
11. PA (Password Attribute)
12. Levenshtein distance (The specific type of edit distance, Google to find more) (Find a library from javascript instead of doing everything on your own)
13. Results of words stored in an array