**Roll Call  
Prompt and Format automation**

**1. Project Description:**Roll Call is a tailor made roll call and formatting program that I created for my wife to use in her professional work. The program iterates through a CSV file and prompts the user to assign a status of attending, apologising, chairing, or taking minutes, to names in the CSV file and then prints them in the desired format. This program has 2 different modes, ‘Rapid Fire’, and ‘Search’. ‘Rapid Fire’ iterates through the CSV file names one by one and prompts the user to assign a status to each one. The ‘Search’ mode prompts the user to type in names for each status type, and if the names match a name in the CSV file, the name will be appended to the selected status list. Both programs have the ability to restart, exit or print at any time.  
  
The purpose of this project was to gain hands on experience using python in a self-guided environment within the context of meeting user requirements.   
 **2. Problem Statement:**As this was my first computer science project, the goal was to solve my lack of experience coding in a self-guided environment. At the time my wife was confronted with a large amount of work, this provided an opportunity to gain hands on experience programming in a self-guided environment to meet user specifications.  
  
**3. Solution Implementation:**  
The consultation was casual and came naturally as I had newly acquired Python skills and my wifes workload was building up. After hearing her user problems and needs, I asked further questions to gather information that would guide my approach, and I applied the solution as follows:  
1. Imported Re(REGEX), CSV and Time libraries   
2. Used input() to prompt the user to select the mode ‘Rapid Fire’ or ‘Search’, and assigned their response to a variable called ‘intro’  
3. Created an ‘if’ statement about ‘intro’, so when the user enters ‘Rapid Fire’ or ‘Search’, the appropriate mode is selected  
4. Included the lower() function to remove capitalisation, and the strip() function to remove leading and trailing white spaces, to account for every variation of ‘Rapid Fire’ or ‘Search’ the user might enter  
5. Created 4 lists that represent each status of attendance (Attended, Chair, Minutes, Apologies) that names in the CVS can be assigned to  
6. Using the print statement, I instructed the user on how to assign names to a list and print the list upon completion  
7. With open(), I opened the csv file as a variable called ‘file’  
8. Using csv.reader(), I read the file and assigned it to a variable called ‘myfile’  
**The Rapid Fire Mode:**  
9. Then I created a for loop to iterate through every name in the file  
10. Within that loop, I created a series of ‘if’ statements with the append function, to respond to the users inputs allowing them to assign names to a status, print the names in their given list, and to restart if an incorrect input is entered   
11. The code above is encapsulated in an object called ‘rapid()’, which is called to restart the program upon false input

**The Search Mode**  
12. Used input() to prompt the user who chaired and minutest the meeting  
13. Created a nested for loop to iterate through the names in the file  
14. Used the search() function from regex to match what the user inputs with the names in the file and assigned it to a variable called ‘match’ (the chair) and ‘match2’ (the minutest)  
15. Embedded in that loop, I created an ‘if’ statement about the matches and used append() to append the match to the corresponding lists  
16. Next I use a while loop to enable the user unlimited searches for the Attendees and Apologies

17. I used the same method as steps (13), (14) and (15) to search the file and append to the lists, but assigned to different variables  
**Both Versions**  
18. Printing the lists in the users requested format presented challenging due to the data type and it being embedded in ‘if’, ‘with’ and ‘while’ loops. The solution was to create a for loop and iterate through the list and using the join() function to add a comma to each object in the list.  
19. Called the time.sleep() function to give the user ample time with the printed results.  
 **the while loop at the beginning is confusing lmao work it out**

**4. Results and Impact:  
  
  
5. Technical Skills Demonstrated:  
  
  
6. GitHub Repository:**<https://github.com/Davidooj/Projects/blob/main/Roll_Call.py>  
  
  
**7. Key Takeaways:  
sub title:**