I realize that this is not the exact format you wanted for the diary. I always forgot to write down my goal before starting to code and would end up just picking up randomly throughout the day and then writing in here every day or two when I went back to the rubric. The way I tend to handle goals is at the top of my code I always have an ordered list of goals to complete and would just go down the list.

Hours 1 - 4: Looking for peer review on AI generated automated essay extension, AI grammar functions, and Auto predict/suggest functions.
Hours 5 - 7: Narrowing down the scope, range was too wide, fall on replication of IOS and google docs autofill/informed autocorrect
Hours 8 - 9: Explore research into specifically autofill/correct functions
Hours 10 - 12: Built out spell check functions, looked for english word dictionaries
Hour 13: Built separate program to join multiple dictionaries together
Hours 14 - 15: Because we had not found a satisfactory data set yet, started to try and build a function that automatically updates probability as you type. Realize we have a BIG problem, need to actively read a text file, or keystrokes. Don't know how to do either
Hours 16 - 18: Focus on trying to explore active file reading options, find nothing online, get frustrated, start trying some crazy computation heavy bootlegging code to fix the problem… does not work
Hours 19 -20: Very frustrated
Hours 21 - 24: Start trying to find ways to read keystrokes, find obscure library on internet, requires special permission from computer administrator, give it, Thonny throws up a million errors all surrounding apples administrator rules, get exponentially more pissed off
Hours 25 - 26: Find new obscure library, import to Thonny, find out it is only operable on unix, try to find library offshoots that work on MAC OS, they are sketch and do not really work as advertised.
Hour 27: Come "crying" (not actually but emotionally) to Prof. Linderman for help, being a computer wizard he finds the write library within 5 minutes and alleviates my coding depression.
Hours 28 - 29: Exploring the Blessed library and playing around with its implementation in Thonny
Hours 30 - 33: Start implementing Blessed into the code, creating the basic outline with comments and elementary code to build direction for keystrokes
Hours 34 - 35: Build act_add_examples, first attempt seems to work
Hours 36 - 37: Start to work on predict
Hours 38 - 46: (Forgot to journal) Realize predict and act_add_examples only work for my specific examples I am typing, spend stupid amount of time trying to fix the code. (Peter finds final data set)
Hours 46 - 50: Rewriting predict and act_add_examples to remove redundancies and accurately work through data set
Hour 50: Write remove_word after finishing add
Hour 51: Start trying to actually output to a file you can read, get defeated
Hours 52 - 58: (Last day, yes I spent easily over 10 hours hanging my head against a metaphorical wall because I am stubborn). Find the correct solution but start to realize complex bugs in the outwriting process, which actually turn out to be larger hidden bugs in our overall code. Very sad. Spent way too long debugging with print statements to try and find errors.
Hours 59 - 62: Clear up as many errors as possible, write comments for code (make it readable), review poster and abstract.
Hours ???: NOPE!!! WE ARE DONE!