

Review Document for Version 1

Silvia Alberti

College of Design, Iowa State University

HCI 5840

Professor Harding

Jul 6, 2025

Overview of the Project

My project, “Step-by-Step,” is a CLI Python program designed to help early childhood educators log and track children’s developmental milestones. The user (teacher) can log a child’s progress by category and visualize the full data frame table, as well as a category line graph and the overall category proportions with a pie chart.

What I Have Implemented So Far

So far, I have completed the “skeleton” of my program. I started by adding a fictional CSV with some data and created the CLI input to log new milestones. I then created a visualization mechanism that uses the data in the CSV along with the newly added logs to generate two types of graphs: the category one (line graph) and the overall distribution (pie chart). I spent my last session cleaning up anything that was redundant (such as the duplicate date column) and added more info to the line graph, such as a title, axis labels, and a simple average function.

Demonstration Video

<https://iastate.box.com/s/uet1imas9hn0y6kyi2d8uiqbdcybqv27>

Issues Encountered

I did not have any major issues. However, I had to do some small fixes as I went, such as replacing `pd.append()` with `pd.concat()`, fixing some errors in the date parsing, and lastly, when I was creating the flag column for the “regression” logs, I had to change the

logic a little bit because, since I did not have it in the CSV, it was showing those logs as NaN. As of today, they are all fixed, so I am not currently stuck on anything.

Where Is Help Needed?

So far, I have had a smooth-ish ride, so I hope not to have any major problems. However, if possible, I could use some suggestions for the things I want to implement in version two. Mainly, I would like to add a nice GUI: do you recommend using Tkinter? Additionally, before I add the GUI, should I think of creating some sort of user input validation? For example, what if the user inserts a letter and not a number? Lastly, is there anything else I need to take care of before creating the GUI? Thanks in advance!

Milestones for the Next Weeks

Below are my ideal milestones for the next steps:

1. **Clean Up and Prep the Current Code:** The first step is to clean up anything that might result in a flaw before creating the GUI. For example, if user input validation is needed, then I will need to troubleshoot that and anything else that might need attention to “prep” the code for a GUI.
2. **Create Lo-Fi GUI to Test Functionality:** Then I plan on creating a lo-fi GUI, only to test if everything works correctly (this might be the longest milestone, so I will make sure to dedicate more time to it).

— *Progress Report A*

3. **Turn Lo-Fi into Hi-Fi (to the best of my abilities):** I will then perfect the GUI and take care of the aesthetics, trying to make it intuitive and user-friendly.

— *Progress Report B*

4. **User Test with Target Users:** Lastly, if I have time, I would like to test the code with different people who are not familiar with the computer science & design field. It would be nice to have teachers do so.
5. **Refine and Finish** (if in time for this class): Use the user feedback to make any final changes before the code is finished and “good to go.”

— *Final Report*

Self-Reflection

I am very satisfied with what I have been accomplishing so far. I am confident that I will be able to finish the project if I follow the milestones without encountering any major issues. I think I was more so scared of Python, but I now realize that with some time and dedication, it is only another tool to learn, nothing terrifying.

As for the light bulb moment, I think that the biggest lesson I took was that I need to treat Python and coding just like math. I cannot imagine crushing a big load of work all at once because it requires a lot of logic and attention. Just like mathematics, it is best to make frequent changes in smaller time blocks.

Also, I've learned that my brain needs to be “rested and ready” to code before I start anything. Forcing my brain to work was not a doable thing, and therefore a lot of times I had

to recognize when it was maybe best to take a break and come back with fresh eyes, rather than getting stuck on a simple issue for hours.