Shannon Buchanan
IS 452
15 December 2019

Final Project Narrative

Original Proposal and Evolution

Not too long after starting on my final project, did I discover that I was being overly ambitious with my abilities. Particularly since my personality does not allow me to submit incomplete work. My original proposal involved creating a database for my personal library collection. Not realizing I would struggle enough with that as is, I planned on adding a web-scrapping element that would pull the metadata from the internet and populate pre-determined metadata fields. During my check-in I revealed my failure to come anywhere close to a complete database, let alone one that can web-scrape. I faced the problem of being infinitely stuck in a loop, that despite hours of research and minor tweaks to the code, I failed to exit. At this point I realized I needed to set my expectations lower, because any programming project that functions properly with little outside intervention will be a success.

Current Goals

Since this revelation, my goal, that I believe I have achieved, is to create a database of sorts that uses user input information to create a file containing one's personal library. My python code can be found here: https://github.com/slb6-final/PersonalLibrary . The python file "slb6_FinalSubmission.py" contains the completed code that, ideally, is debugged and runs properly. The program is divided into three functions, with another minor function used to determine whether an input is True or False.

Main Function

When run, and after introducing itself, the program prompts the user with a question, would they like to add an entry to the library, if the answer is yes, the add_entry function is executed. If the user answers no, a second question is asked as to whether they would like to remove an entry. If they answer yes, the remove_entry function is executed, or if they answer no, the statement "You have finished your personal library file." In order to easily allow for multiple forms of 'yes' and 'no' to be used, a function titled yes_no is used. This function has sets that contain variations of the words 'yes' and 'no'. An if statement is then used to determine whether the user input answer falls within the 'yes', 'no', or 'other' category. If the answer resides in the 'yes' set, then a True condition is returned, if the answer resides in the 'no' set, then a False condition is returned. If the answer falls outside of those two sets, then the user is faced with the message "Please respond with 'yes' or 'no'". Currently a means of reentering the if statement is not written, so the program moves on without an answer if not put in properly.

Add Entry Function

The add_entry function has been the bane of my existence for a few weeks. I spent hours attempting to remove myself from the endless loop created from the while statement. I knew I just needed to figure this out instead of re-write it another day, because I wanted the user to be able to input as many entries as they would like at a time. After dedicating five straight hours to tweaking and research, I eventually found my way out, but do not ask me how I did it, because It was a blur. Over the course of this project I went back and forth about creating a csv file, or a txt file, or a json file. After trying and failing at all three, I eventually went back to the csv file because I had an epiphany, or at least found a website that answered all my problems. The main issue I was facing relates to removing entries, but I will get into

that later. In the add_entry function I played around with adding metadata to dictionaries that would then be appended to a list, but when I went back to the csv file, I completely scrapped that idea. The commented-out code for this can still be seen in the file: “slb6_Final_CheckIn_2.py”. When I reverted back to using a csv file, I used a similar format as we have done when writing csv files in class. I created all my variables, had user-input answers set to them, and then wrote the whole thing to the file. The standard metadata fields I have for entries are title, author, ISBN, copyright date, and publisher. After those entries I have a question as to whether the book is in a series, if so the series name and the place the book falls in the series are required to be answered, if not blank fields are entered. After all the metadata has been input for each individual entry, the information is then written to the csv file. A prompt asking whether the user would like to input more entries is then asked. If a true statement is given, the loop starts over, if a false statement is given, the loop exits and returns to the main function.

Remove Entry Function

The remove_entry function dictated which file type I would use from the beginning. When I first attempted it with a csv file, the entire library would be wiped when I attempted to remove a single entry. When I attempted with txt and json files, I could not even figure out how to display the entire library on the console. After much trial and error, and copious amounts of internet research, I discovered a simplistic way to use csv files to remove the entries. This involves reading the file into a list, removing the row that contains the title that the user intends to delete, and the re-writing the entire list to the file. I honestly wish I had discovered this sooner because of how easy it is.

The Future

If I had more time to work on this program, as it currently stands, I would strive to debug it further. For example, I would prompt the yes_no function to restart if an invalid entry was input. Additionally, I would add a loop to the remove entry function so that a user may remove multiple entries without restarting the entire program. Lastly, I would add on the ability to view the library, in its current state, from the main function, so that the user is able see what is listed and what may be improperly written out. Much further into the future, I would like to add in search capabilities somehow from the main function, as well as adding a user interface.