CSCE 240 Final Project

Marcus Leano

This final project is a culmination of all our previous projects, from PA1 to PA5, incorporating elements and feedback from each project to create an overall chatbot that could answer user queries, predict user queries from their given input, store chat sessions, and report on previously stored chat sessions. This all revolved around the 10-K reports of two individually chosen companies, of which I chose both Paramount Global and Netflix, two leading media companies in streaming. Overall, this project has helped me learn more about C++ and its overall coding structure, how to structure classes, and more on object oriented programming.

Throughout this semester, I decided to code this project using C++. I wanted to use C++ because I had no prior experience of using C++, and it would help me to follow along with the concepts I had learned in class. For unfamiliar concepts that I did not know how to tackle, I utilized forums and codes online that I could learn and take from, as well as correctly citing and crediting the work if I decided to implement it into my program. I spent a decent majority of my time learning concepts and finding solutions to bugs and problems in my code to better understand C++.

The scope of this project mainly revolved around implementing the different requirements, and trying to ensure code that was easily readable and reusable. For the most part, I went with more simple solutions due to my lack of experience, but I attempted to use headers and classes to help make the code less cluttered. Therefore much of my code did not attempt to go beyond the requirements, just build upon what I already have.

PA1

The first part of the project involved extracting the company 10-K files from the website and parsing their information in a simple manner. I was unfamiliar with getting a scraper to extract the data directly from the website, so I opted to download the 10-K files as a text file and use that information. However, little things like page numbers were present in the file, littering the text file with filler lines. I opted to keep them in the file for now, but I would use a method in future PAs to clean up these filler lines when I presented information to the user.

PA2

The second part of the project involved extracting items and parts from the 10-K files. Overall, using regex to search for specific parts of the file for the program to start and stop reading made this relatively easy. I also implemented a method to read the file for any filler lines and skip those being outputted to the console so the user was not given cluttered information. In addition, I noticed reading the main section of certain parts would just tell the reader to look at a different section of the file, so I created separate methods for these sections to find the redirected parts of the file and read from there.

PA3

For the most part, I had already created most of the requirements for this project in the previous part, as my project ran in an infinite loop until the user inputted ‘quit’. However, I did add more complicated regex so that the program could recognize common mistakes and misspellings. In addition, I had trouble with the previous project in attempting to implement headers and classes, which made my main cpp file cluttered and be much longer than it is now, so I learned how to correctly implement classes so that I could put my extractor in a separate class.

PA4

This part of the project was the one I had the most trouble with, as I was unsure on how to implement an intent to query mapper that would have been more effective than regex. All solutions I thought up felt like it would not be as effective compared to regex, so I spent time refining my regex and creating methods that could parse user input word by word and determine whether it should read from the Netflix or Paramount file based on how many times the user uttered the company's name over another.

PA5

I found this part of the project the most fun, and I enjoyed figuring out how to save the statistics and chat logs in different files. Some of my code was not very optimized, so I borrowed code from Charles Grant's, particularly the method that read the logged chat from a text file, as it felt more efficient and easier to read than what I had. I had little problems in implementing these methods as his code was easy to parse and modify to fit my program. Grant had also borrowed some of my PA5 code as well as there were some methods I made that felt easier for him to implement than what he currently had.

Finally, after performing some small changes, I had integrated all the requirements from previous projects into one program that could handle most of the requirements relatively well. I'm a bit disappointed that I did not create as complex of an intent to query mapper as some students in the class had, and it is definitely something I would want to take a second look at even after the end of this project.

Overall, I felt like I had done a decent job at the project in spite of my lack of experience, and I had learned a lot in regards to creating classes in C++, incorporating knowledge from my

previous experience with Java, and overall more on how C++ works as a programming language and how it differs from Java. There are definitely some parts of the project that I could improve on, such as the requirements for PA4 and optimizing the code. Future implementations of the project using my code would definitely need some improvement in some areas to make it more reusable, and there are some parts of the code that are specific to my companies. In addition, a way to extract the information from the website itself instead of having to manually download the files and put them in the respective folder would allow my program to be more user-friendly and allow easier implementations of other companies. I hope to take what I learned here in this project and expand on it so that my proficiency with the C++ language is more refined.