

Chatbot Sessions Links:

- <https://chatgpt.com/share/68f1255e-d5e0-800e-9ce9-619cc2a6c33f>

Chatbots used:

I chose to use ChatGPT for the entire assignment because I am most familiar with it.

Input file generation

- I had the chatbot generate all the input files. I prompted it to create normal and input files with mistakes such as invalid formats, missing information or invalid ratings.

Incorporating issues in input for correctness testing.

- I used ChatGPT to incorporate issues in the input files for testing. Additional issues added are negative ratings, duplicate movie titles, and missing user or unknown movies.

Ensuring good distribution of input data

- ChatGPT initially returned a test code that would check a few of the edge cases but not all of them. It missed things like empty files, non-existent files and partial data. After prompting ChatGPT to include additional tests, it successfully added the code to create input files and test missed cases. Originally, it tested each case without showing the actual outputs. I changed it to include outputs so it was easier to double check on my end.
-

[40 pts] Documentation - Testing

Clearly document the following, in a file named testing_doc.pdf (again, only PDF accepted)

- **[5 pts] Input file generation**
Did you generate input files? Did you have the chatbot generate input files? Detail which files were generated by you, and which ones by the chatbot.
 - **[20 pts] Incorporating issues in input for correctness testing**
How did you incorporate possible issues in the input files so you can test the correctness of code that handles these issues?
 - **[15 pts] Ensuring good distribution of input data**
How did you ensure a good distribution of data in the input files so that you could be confident that your tests exercised various paths through the code?
-