The beta release document of the twitch vod visualization highlights the design and development work done by the team. The design of the website is explained quite well and the upgrades before the final release are understandable. The beta release document should have included the high-level understanding of the project to give the client/tester a clear picture of the goals accomplished by the functionality. A description of the problem statement is a necessity for clients with non-technical backgrounds.

The website functionality as described in the document seems to be working correctly on checking with twitch vods. The API also does retrieve the YouTube vods and give the chat JSON which can be downloaded and further analyzed. Hence the core features of ingesting user input, querying the API and dynamically creating visualizations from the VOD chat are fully functional. The visualizations are insightful and give a good understanding of the VOD's popularity and interaction. There are minor bugs in the visualizations which can be easily corrected before the final release. For the first 6-line charts, the x axis can be changed to 30 sec intervals instead of 1 unit representing 30 secs, and include the label for x-axis. The number of messages per user chart can be reduced to show only a handful of interesting users instead of everyone (or < 15 users as indicated in the Bugs section) who messaged (for better visual representation). The emote usage chart also could be modified to efficiently represent the outliers, as the chart that was represented for my input only showed one prominent value and the rest of the values were near zero.

The report does show ambiguity with respect to implementation of cache to prevent overloading the API. The document addresses that it is built on the client's end and the team is deciding whether to implement it on the server's end. I believe this is a decision that needs to be done prior to the beta release and if not decided yet, to drop that functionality altogether. The resizing of the visualizations was tested and is well received. Although the error codes do their job, they can be extended to provide a sample correct input. The error codes should be handled better so that a non-technical person can understand the error and correct the input. The YouTube URLs functionality is one that should be dropped in my opinion as it would take considerable development and testing time. But if the development is small enough to enable an extension, that should be ideal. However, the testing for YouTube VODs should be rigorous

The statement under the 'What we accomplished' section in page 1 talks about hosting website and API on 2 cloud providers, and that 'grants access to our app from any interested users'- this statement is very vague and possibly mis-typed which could be changed for the final paper. Apart from this, the document is very clearly written and grammatically sound.

I faced difficulty in reproducing the visualizations as the input URL I was entering didn't have the same syntax as the developers intended it to be. It was hard to figure out the error codes as to what was going wrong. A simple input example embedded in the website error message or in the input field itself would've sufficed. This can be implemented before the final release and the input specifications can be described in the final paper as well.

The bugs and enhancements are well received and indicate a strong understanding of the goals to be achieved before the final release. With some modifications, the final release looks all clear and set! I would give an aggregate score of 17 with the division as follows:

1. Functionality: 4 (-1 due to YouTube's non-implementation in the beta release)
2. Report Spelling and Grammar: 5
3. Reproducing functionality at Client End: 3 (-2 due to errors when URLs are input and insufficient error handling mechanism to indicate what went wrong/sample inputs)
4. Bugs & Enhancements: 5

The UI was interactive and is easy to use. The representation of the summary stats at the webpage header is good. I believe the interactive resizing to accommodate the mobile phone users is great and well implemented. I believe the team has all the answers of how to improve before the final release and must implement them. The most important suggestion would be to handle the inputs better by error messages or simple sample inputs.

I could find a twitch chat implementation using d3.js and angular on github. The study dealt with searching and analyzing the main scene of the video using the video and chats. https://github.com/chaht01/twitch-chat-visualization