Project Progress Report

1) **Which tasks have been completed?**

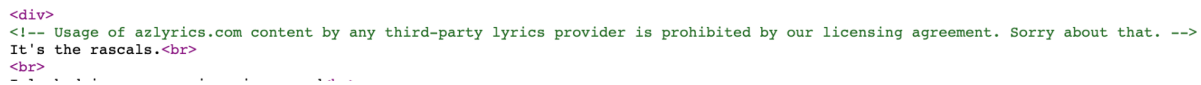
- Started the UI of our web application in ReactJS. We've started integrating the Spotify API into the website to display song cover art as clickable links.
- Begun constructing the song database in a CSV file.
- Experimented with the scikit-learn semantic analysis library in Jupyter Notebooks with Python to see how well the library analyzes different phrases and discovers the overall sentiment/mood of the phrase.

2) **Which tasks are pending?**

- Fine-tune our song database
- Create a local backend server using FastAPI
- Create our Python sentiment analysis function to analyze the user-entered phrase and return a list of songs that match the sentiment of this phrase to the user interface
- Display song playlists in the user interface and finish the design of the website
- After all these steps, we will implement the user feedback buttons in the UI and create a feedback function

3) **Are you facing any challenges?**

- We were trying to figure out which lyrics API we wanted to use for the project and started with AZLyrics. The API worked well and we were able to fetch songs easily, but after lots of experimenting with this API, we unfortunately realized the site had a legal statement that said:



  We do not want to risk any legal issues with this API, so we had to look for different ones. We are now planning to use the Genius API.
- Another challenge we're facing is whether to call the SpotifyAPI or store the audio files for the returned song playlist in our local servers. We need to experiment with both options to see which one performs better.
- It is also tricky to find the "most" accurate sentiment analysis library in Python (scikit-learn vs. BERT vs. PyTorch) for our specific task.