

## Team Member Names and Emails

- Aishik Ghosh [aishikg2@illinois.edu](mailto:aishikg2@illinois.edu)
- Alvina Waseem [awaseem2@illinois.edu](mailto:awaseem2@illinois.edu)
- Elaine Wang [elainew2@illinois.edu](mailto:elainew2@illinois.edu)
- Isha Kukadia [kukadia2@illinois.edu](mailto:kukadia2@illinois.edu)
- Sarah Nadeem [snadeem2@illinois.edu](mailto:snadeem2@illinois.edu)
- Suhirtha Raj [sraj6@illinois.edu](mailto:sraj6@illinois.edu)

## Final Project Documentation

The primary function of our web application is to analyze Facebook Messenger group chats for dynamics. Many of the features of our app are centered around reactions (i.e., like, love, haha, wow, sad, angry) that Facebook allows. For instance, the app allows users to examine group member dynamics by calculating who sends and receives the most reacts in the group and compartmentalizing the reacts into what reacts each participant in the group sends and receives and who they send and receive reacts from. This functionality is achieved by parsing the JSON of the chat that Messenger provides using Python. Furthermore, the app can be used to inspect react statistics associated with specific messages in order to collate messages with at least a certain amount of reacts and metadata associated with those reacts. This data is enhanced using sentiment analysis to see how participants react to messages.

Our web app also performs basic natural language processing by utilizing the Natural Language Toolkit. The app employs the corpus provided by NLTK in order to determine the most common misspelled words in that chat. If a word is misspelled often enough, it is considered to be part of the chat's slang. Stop words are not considered during this analysis. The NLTK library is also used to determine the most and least common words used throughout the history of the chat. The code is multithreaded since NLTK is a heavy library and removing all the stop words when analyzing a sufficiently long chat is arduous and time-consuming.

In addition to these features, the app also provides basic metadata about the group chat that users might be interested in. A non-exhaustive list of these features includes who makes the most changes to the chat (i.e., sets nicknames, starts polls, etc.), who has the most mentions in the chat, and whether users can be classified as night owls or early birds. This is accomplished by parsing the chat JSON and comparing the sentiment of the chat before and after the changes were made.

The data returned by these functions is displayed on the app using the Vega visualization grammar. This library utilizes the JSONs outputted by the previous functions and generates graphs and art in order to interpret the data generated by the app. The front-end is created using HTML and JS. No additional software needs to be installed or run in order to view the visualizations. Users may simply visit the URL and submit their group chat JSON in order to analyze their chat.

Sarah (Irdina) Nadeem, Suhirtha Raj, and Elaine Wang analyzed the changes made to the chat in addition to performing data visualizations and deploying the visualizations to the web app. Aishik Ghosh and Alvina Waseem performed text analysis on the chat using NLP libraries. Isha Kukadia was in charge of the react statistics and sentiment analysis associated with the reacts.