1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.
   Nikitha Nagumalli - nikitha4 (captain)
   Moksh Shah - moksh2
   Ishan Nikam - inikam2

2. What is your free topic? Please give a detailed description. What is the task? Why is it important or interesting? What is your planned approach? What tools, systems or datasets are involved? What is the expected outcome? How are you going to evaluate your work?
   Our idea for the project is to build a tool that allows users to search for text in their facebook messenger history and ranks which of their contacts are most likely to contain the query in the corresponding messages. In order to do this, we will have the user download their messenger history, upload it to our tool, and then analyze the transcripts by looking for the user's query in the transcripts and see which contact's conversation is most likely to contain that query. Specifically, we are going to do this by keeping a bag of words representing the messenger history for each contact, calculate the probability that the query is within each of these messenger histories, and output the rankings of the contacts based on their corresponding probabilities. We are going to evaluate our work by uploading a small dataset of messenger history that we will manually figure out what the output should be for a set of queries and then check that our tool's output matches what we expect our output to be.

3. Which programming language do you plan to use?
   We plan to use python for the backend code. This is where all of the data processing and logic for calculating probabilities of the query existing in each contact's messenger history will live. For the frontend, where the user will input a query and the results will be displayed, we are planning on using javascript and html.

4. justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

   - Learn python - 3 hours
   - Learn javascript - 3 hours
   - Learn html - 2 hours
   - Learn how to connect backend and frontend - 4 hours
   - Figure out how to get messenger data - 2 hours
   - Pre-process data to keep useful information - 3 hours
   - Process message data into inverted index - 8 hours
   - Implement ranking function for determining who is most likely to message the query:
     - Use max likelihood model to rank users (6 hours)
     - Implement smoothing to ensure non-zero probabilities (2 hours)

    - Integrate with front-end (4 hours)
- Search function displaying the messages in which the query occurs, ranked from most recent to oldest - 6 hours
- Building Front-End Tool -
    - process query - 2 hours
    - display output - 2 hours
    - styling - 1 hour
- Documentation -
    - readme - 1 hour
    - explain code in comments - 3 hour
- Testing -
    - Make fake evaluation data (messages and queries) - 1 hour
    - Manually compute output - 2 hours
    - Check against tool's output - 1 hour
- Progress Report - 1 hour
- Final Demo -
    - script - 1 hour
    - video - 2 hours