# Team Name - Intelligent Analyzers

Group Members:
1. Atharv Chandratre - atharvc2@illinois.edu (Captain)
2. Uday Kanth Reddy Kakarla - uk3@illinois.edu
3. Priyanka Awatramani - pma7@illinois.edu
4. Ansh Bilimoria - amb20@illinois.edu

## Progress Made Thus Far:

As a part of this project, we have been able to complete the first two tasks of our proposal. These two tasks were:

1. Scraping of online data in order to get information about the companies we are trying to gain insights and sentiments about.
2. Pre-processing of the data we have collected in the first step. This consisted of the completion of the following tasks on the data:
    a. Tokenizing the textual data into tokens which can then be processed further.
    b. Removal of the stop words (as we have learned in the course) from the tokenized data.
    c. Removal of punctuation signs and other unwanted/non-contributory tokens which the tokenized data contained.

We leveraged NLTK (Natural Language Toolkit), a Python-based Natural Language Processing library to complete the above tasks. NLTK comes out of the box with functionality for some of the tasks mentioned above, and utilizing this powerful library for the project enabled us to make progress on the tasks. We are also using Jupyter Notebooks on Google Colab for running the computations on the data, and this is turning out to be a good choice as Google Colab's processing power is quite high, enabling us to perform computation and scraping with decent speeds.

Currently, we are making progress on our third task as written in the proposal - which is to take a lexicon-based approach and run the pre-processed information through a sentiment lexicon.

## Remaining Tasks:

1. We are looking forward to accounting for Part-of-Speech tagging as part of preprocessing the data as this could give us a better sentiment result. We also intend to take into account emojis and hashtags as part of the sentiment analysis and not just words.
2. We haven't yet completed the lexicon-based approach and we are trying out different ways to see which method can give us a more accurate sentiment score. One of the ways would be using a company-based dictionary which has some work-related slang words and acronyms that are not present in the original English dictionary and thus classifying these words as positive or negative would improve our model.
3. Lastly, we are yet to give recommendations based on the sentiments predicted by our lexicon-based model. These suggestions would help teams to improve communication and their job experience.

## Challenges/Issues We Faced/Overcame:

1. While scraping the data from Reddit (r/CSMajors), there were a couple of reviews that didn't actually contain the name of the company. Hence, it was difficult to decide which company a particular review belonged to. For example, a lot of times, the companies in question were referred to by their acronyms (MSFT for Microsoft), or by an offering of the company (AWS for Amazon). So if a review was about Amazon, but only used the word AWS in the review, the parser would not know that it was referring to Amazon. We understand they are the same because of prior knowledge of the world, that AWS is a team within Amazon. For this reason, we decided to scrape data from Indeed.com, as the data seemed to be easier to scrape and more structured.
2. After deciding to move to scrape Indeed.com, we realized something else. As new reviews keep getting posted, we need to keep scraping the company's reviews page to get an updated idea of employee satisfaction in the company. This is a challenge, as we need to decide at what intervals we want to send a request for getting updated data about a particular company. A similar case occurs when trying to keep up with the structural changes of the website. Even a minor change to the UI of the website might require rewriting significant parts of the parser from our end. The web scraper we implemented was specifically written with respect to the code elements of Indeed.com at the point of setup, so frequent changes complicated the code.
3. Indeed.com may be slow to load content or may not load at all when receiving a large number of access requests. In such a situation, we need to refresh the

page and wait for the site to recover. However, our parser will not know how to handle such a situation and data collection can get interrupted.

4. Passing the login credentials for creating a cookie value addition posed a challenge as we had to create a new sort of test user to be passed to our scraper configuration to get access to the content and start the scraping process.