Problem Statement:
You are tasked with building a data store of misinformation and hate speech related content and then allow social media platforms to match against this content. Select a self-defined piece of the problem, describe your solution and write some prototype code for it. You can pick whichever part of the code you are most comfortable with (frontend, backend, DS/ML or system) and use your own sample data if you like (not required).

This is a challenging task with considerable ambiguity. Rather than aiming for completeness, solve a small but well defined sub-problem. For example, each of these is a valid subproblem, but there's many more that you can think of as well:

1. Create a way for users to report misinformation and hate speech content to trustlab, so we can add it to the data store. Considerations: How can you make it as easy as possible for non-technical users to report such content? What information is useful to gather so the reported content can easily be matched by social media platforms?
2. Create a scalable data store to store misinformation and hate speech content. Considerations: What kind of data store to use? What format/schema to store data in so we can easily match against other other content from social media platforms? How to make the system efficient in terms of storage and/or performance?
3. Create a way to build high quality ground truth. Note users will not self-report accurately. (E.g., "I hate the governor" isn't hate speech.) Considerations: How to handle ambiguity in what's considered misinformation or hate speech? How to reduce resources needed, e.g., human resources? How to handle cultural issues?
4. Create a classifier to determine whether a piece of content is misinformation or hate speech, assuming you have already collected ground truth. Considerations: What more can be done beyond leveraging standard off-the-shelf techniques (e.g., BERT)? What makes this solution approach unique vs what other companies might be doing?
5. Design appropriate interfaces/APIs for social media platforms to utilize the data store. Considerations: How to make the API/interfaces cover different use cases so it's easier for various platforms to integrate it within the context of (different and varied) workflows they might have for handling content uploaded to them by their users.

Each of these sub-problems itself is broad, but there are other ways to further restrict the scope. E.g., only handle misinformation or hate speech but not both, or only just one type of misinformation or hate speech. On the content side, you can choose to restrict to comments only, or social media posts only, text content only, or video content only etc. On the platform side, you can choose to restrict to specific platform(s), e.g., Facebook or Youtube or Tiktok or blogs or news sites or discussion forums. Accordingly, feel free to make reasonable assumptions about the format of the content. On a similar note, given you are only solving a subproblem, you may assume that other subsystems exist for other parts of the problem that your subproblem depends upon, and where necessary declare reasonable interfaces with placeholder/empty implementations for these other subsystems.

Please note:

- Don’t feel obliged to restrict yourself to industry standard practices, use your imagination and creativity. This is essential for the real job too because innovation is at the core of solving the problem better than it has been solved so far.
- You may want to plan your time carefully. Spending 4 hours defining the sub-problem will not leave you enough time to produce working code. On the other hand, jumping to the first sub-problem that comes to mind may not be wise either.
- Feel free to research online, refer to textbooks, leverage packages/libraries etc, but work independently, as you will on the real job.
- We understand you may have a lot of questions as you approach this, but on the flipside you get to make your own rules. Imagine that you are starting up on your own, and you are your own boss!

Coding related instructions:

- Please create a project on GitHub under your own name and add TLinternship2020 as a collaborator so we can view your work after you submit.
- Please do not upload executables, just source files.
- Please include a README that describes the sub-problem, the solution approach, any libraries/packages you are using as dependencies, and how to run your code.
- Feel free to use any language/platform that makes you effective and you have access to (e.g., AWS). If you have a choice, we prefer python, javascript/typescript and go.

Good luck and looking forward to hearing from you in 6 hours latest :)