# Software Engineer - Technical Challenge

## Instructions

- There is no strict time box for this challenge, but please do not spend days on it. We are looking for a demonstration of what you can do.
- If you are unable to finish, then please also leave some thoughts on how you would finish the solution. You may also wish to leave comments about how you would improve your solution in general.
- We are looking to see if you can implement code that is well structured, clear, concise, tested, and easily understood by others.
- Feel free to reach out to us with any questions you might have or any clarifications you need.
- We have purposely not defined exactly what the solution should be. Feel free to get creative and think what might be the best way to solve the problem.
- Please make sure that your solution will build and run, and provide instructions on how to do so inside a README.

## The Challenge

Write a simple web application in the frontend technology of your choice which serves a single page where a user can paste a list of URLs in a textbox input and submit them. An example of this list is provided in the text file called “urls.txt”.

The web application must call an API endpoint in the Python application described below, which will process the submitted data according to the requirements below and return a result.

The web application must display the results from the API call to the user in a readable format.

The Python application must satisfy the following requirements.

- The application must run an HTTP API service with one POST endpoint (`/urlcounts`)
- The POST body data should be valid JSON.
- When the endpoint is called, the application must read data into memory from a text file called “hosts.txt” which will contain a list of hosts (e.g. youtube.com, thepiratebay.org, etc).
- The application must deduplicate the URLs that were POSTed to ensure that only unique URLs are counted.
- For each unique URL:

    - Determine whether or not the host (domain name) portion of the URL matches a host found in the hosts text file. Keep a count of how many URL’s match a host and how many don’t match any hosts.
    - For each host, keep a count of how many matching URLs are submitted, regardless of whether there is a match or not.
- The endpoint should return a valid JSON response containing the following:
    - The count of URLs that did match a host
    - The count of URLs that did not match any hosts
    - A list of all unique hosts for which there was a matching URL with a count of URLs that matched
    - A list of all URLs that did not match any hosts

## Submission

Please create a private GitHub repository for your solution and share it with ryan.tromp@corsearch.com once you have finished.