Your goal is to create a demo search engine using:

- Rust
- HTML/CSS
- AWS Lambda

To source the results you will use:

- Bing search API (https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api/)
- Google custom search API in 'entire web' mode (https://developers.google.com/custom-search/v1/overview)

The search results page should look similar to this:
https://engine.presearch.org/search?q=why%20duckduckgo%20sucks%3F

Feature 1) In order to create your own version of top 10 results, you will get 10 results from Bing and 10 from Google. Then you should combine them in any way you think is best.

No other results will be available to the user (no pagination). Also no images, videos, news results for this demo.

On top of the results you should display time to get results from Bing, Google and total time to render your page (in ms).

Feature 2) For each of the 10 results, you will create your own custom text snippet! This means you will process the result page and extract what you think is the most relevant text to show for the query. Also you will bold keywords as you see fit. While generating your snippet you must not read, access or in any way use snippets provided by search engines (no peeking). When hovering above your snippet, you will show the original snippet from Bing or Google. This entire feature should be an option (checkbox on/off; default off) in the interface. If off, you will show search snippet as provided by search engine.

Feature 3) If the user query starts with 'weather' 'stock' or 'time' you will render a widget on top of search results showing weather, stock or time information.

For example 'weather san francisco' should show weather in San Francisco. 'stock aapl' should show Apple stock information (only major US stocks). 'time tokio' should show current time in Tokio.

To render the widget you can use any API you want but you can not embed a ready made widget - you have to create it yourself from the API result.

That's it! You will deploy the code on AWS Lambda.

Deliverables:

- Document explaining overall architecture of the solution, challenges you faced, how you solved them and explanation how you combined the results from Google and Bing and why so
- Access to github repo to evaluate the code
- Link to the demo
- Timesheet showing time you needed to complete the task with break down of individual items

Scoring:

- Most important metric will be performance. This will be measured end-to-end, meaning from the moment users submits the query, to the page is completely rendered. Your goal is to make this as close to possible to search results API time (which is about 400-500ms). For normal results (without snippet generation and special widget) a great result would be +100ms to worst search engine API response (ie if Bing is slowest and it returns API in 300ms, total rendering time should be 400ms + Lambda latency). With snippet generation this will vary depending on sites, but the metric will still be important.
- Next one will be overall code quality, readability
- Third one will be quality of the user facing interface

Good luck! **Feel free** to relay any questions during test via upwork or vprelovac@gmail.com