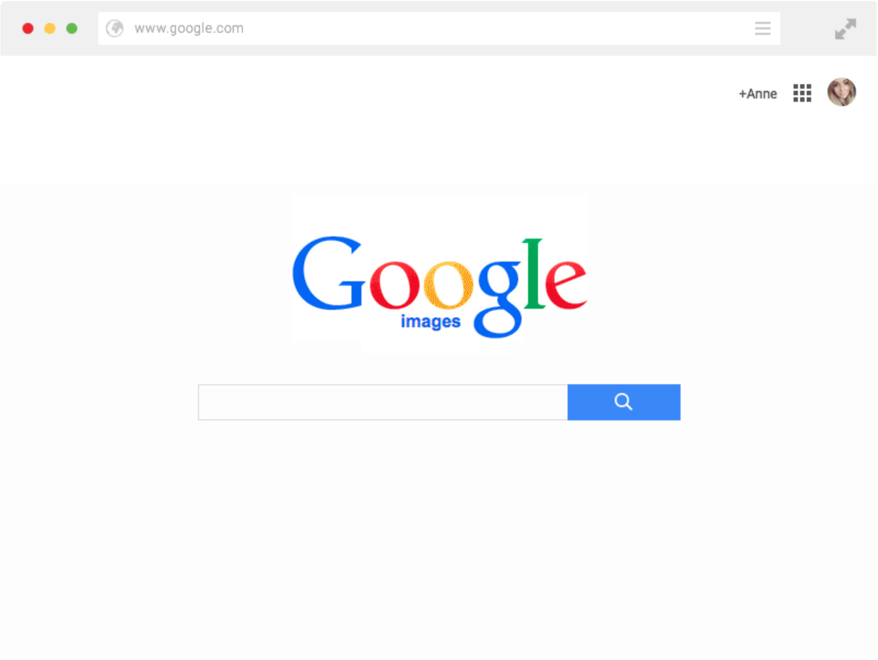
Project

Develop a service that fetches a profile image given an email address.

Client-side

At a minimum, the view should provide an input for the user to enter an email address and button to execute the search. Upon success, the resulting image should be displayed. If there is an error, an appropriate message should be displayed.

While not necessary, your creativity can make you stand out and demonstrate your ability to implement more sophisticated UIs. Here's an example redesign for Google Image search that has a nice transition between the search and the results to draw inspiration from:



source: https://dribbble.com/shots/1987110-Googling-Mad-Max

Server-side

At a minimum, the server should expose a REST API endpoint that accepts an email address as input and responds with an image URL as output. Feel free to use Google, Bing, your other favorite search engine, FullContact, etc. to do the actual query. Any errors should be handled gracefully with some kind of message or notice provided to the user that something went wrong (e.g., no pictures were found, some other error querying your data source, etc.).

The email address results should be cached to expedite subsequent calls. The cache should last at least long enough to test.

NOTE: The accuracy of the results is not important. If I enter my email address and I do not see my picture but instead see some other picture, that is fine.

Please do the following

Create the code in a github repo that you share with user *kristofmic*

Get the code running on a hosted service, so it can be live tested (don't worry about DNS). Both AWS and Google Cloud Engine offer free hosting, but you can use whatever hosting service you like.

Guidelines

Please ask as many clarifying questions as you need! It's what you'd do on the job, right? Send questions to *chourihan@yikyakapp.com*.

There's no specific timeline—please determine how much time you think you'll need, and let *chourihan@yikyakapp.com* know. Again, just like you'd do on the job.

On the client, use whichever libraries or frameworks you are most comfortable with.

On the server, the preferred languages are JavaScript, Go, or PHP. If you would prefer another language, let's talk!

The service is meant to be simple; don't overcomplicate the codebase but architect it as if you would be maintaining it and adding to it over time, so modularity is important.

Have fun with it!