

# SWE/SRE Interview Homework Questions

Programming - please pick any 2 of the 3 questions below. For FE /Fullstack, choose the webapp/ favorite dog exercise and 1 of the other three.

## Webapp - 'favorite dogs'

Create a React webapp that will present a gallery of 6 random dog images from <https://random.dog/woof.json>. Allow user to choose one (or more) to favorite (use local storage).

If user doesn't like any of the six, use a refresh/next button to get 6 more. Repeat.

There should be two endpoints.

/ - described above

/favorites - displays all the favorite dogs

Consider making the screen a flexbox so that it arranges itself correctly for mobile screens.

## Scripting - 'compare versions'

Compare two version numbers version1 and version2.

If version1 > version2 return 1

If version1 < version2 return -1

otherwise return 0

You may assume that the version strings are non-empty and contain only digits and the 'dot' character. The 'dot' character does not represent a decimal point and is used to separate number sequences. For instance '2.5' is not "two and a half" or "half way to version three", it is the fifth second-level revision of the second first-level revision.

Here is an example of version numbers ordering: 0.1 < 1.1 < 1.2 < 1.2.9.9.9.9 < 1.3 < 1.3.4 < 1.10

## Application - 'math api'

Implement a web service (preferably in Go, Python, Ruby or Java; extra effort to do that in Go will be recognised; using a framework or not):

/min - given list of numbers and a quantifier (how many) provides min number(s)  
/max - given list of numbers and a quantifier (how many) provides max number(s)  
/avg - given list of numbers calculates their average  
/median - given list of numbers calculates their median  
/percentile - given list of numbers and quantifier 'q', compute the qth percentile of the list elements

No need to be concerned with resources, we're assuming there's plenty enough of memory, etc.

## **Application - 'tracking web server'**

Implement a small webserver (preferably in Go, Python, Ruby, or Java; extra effort to do that in Go will be recognised):

/ping - returns response code 200 and string OK when file /tmp/ok is present, if file is not present returns 503 service unavailable  
/img - returns a 1x1 gif image, and log the request ~~in apache common log format~~

Server needs to scale to many concurrent users and be efficient. Propose improvements you'd like to work on and - time permits - implement.