Take-home Assignment - Software Engineer

Anvil has deployed an HTTP endpoint at https://dev-assessment.anvil.app/_/api/docs/1 that returns a json payload with a few characters at a time from a text document, and a URL to query to receive the next few characters. For example:

```
{
  "text": “xyz abc”,
  "next_page": "https://dev-assessment.anvil.app/_/api/docs/1?p=2"
}
```

If "next_page" is missing or null, you have reached the end of the document.

We would like you to develop two programs that use this API, as specified below. If you are applying for the Junior Software Developer role, you may write the code in any programming language you are familiar with - as long as it runs! If you are applying for the Software Developer role, please write the code in Javascript, Python or Clojure. Your program will be judged on the assumptions you have made and the quality of your code. Please comment your code minimally, listing any assumptions and non-obvious design and implementation decisions that you make.

This task should take about 1-2 hours. Please email your answers to recruitment@anvil.works with your source code, minimal instructions for how to invoke the program, and an indication of how long you took.

Note: We advise reading both questions before starting.

1) Write an application that takes a URL as a command line argument, reads the document to the end, and outputs a list of word frequencies, ordered by word count and then alphabetically. You should use the starting URL https://dev-assessment.anvil.app/_/api/docs/1 as a test input, and send output to the console.

For example, if the full document reads *It was the best of times, it was the worst of times* then the output will be:
it: 2
of: 2
the: 2
times: 2
was: 2
best: 1
worst: 1

2) Write an application that takes a list of URLs as command line arguments, reads their documents in parallel and produces a console output of combined word counts so far, split by word as in the first step, every 10 seconds. You should test it with the following URLs, which implement the same API with random delays:

https://dev-assessment.anvil.app/_/api/docs/**1**?slow=true
https://dev-assessment.anvil.app/_/api/docs/**2**?slow=true
https://dev-assessment.anvil.app/_/api/docs/**3**?slow=true