DA376E - Lab #3

INSTRUCTIONS

This assignment consists of one task. This is an individual assignment. The lab will be graded pass or fail.

TASK 1 – "BIG DATA" SIMULATION

This lab assume that you have done the previous labs. The labs consist of two parts, the server and the client.

I have provided some sample code with the lab, here:

https://hkr.instructure.com/files/697285/download?download_frd=1

Before running the client and server you need to install the dep, "npm install axios" & "npm install express"

First start the server.js with the command "node server.js" and then run the client "node client.js"

You should get an output:

```
$ node client.js
{ data: 'O Hi this a test Car River Deer Car Bear and' }
0000000000
```

The client does a POST request towards the server that is running on port 3000.

In the Post request the following data is submitted:

"O Hi this a test car River Deer Car Bear and"

In this lab you are going to implement a word length frequency counter.

So, if the following String is submitted "`'O Hi this a test Car River Deer Car Bear and`"

You should respond with "`2 1 3 4 1 0 0 0 0 0`"

This response means the following.

2 words with length 1 (O, a)

1 word with length 2 (Hi)

3 words with length 3 (Car, Car, and)

4 word with length 4 (this, test, Deer, Bear)

1 word with length 5 (River)

There are some sample code to get you started for the backend. The backend only has to support word length of maximum 10.

Also modify the Client so that the result is displayed in a nice table printout or something similar, not just the raw string as in my example.

- Implement word length frequency counter according to description above.
- Deploy the “server” part on Heroku or any other Cloud service provider.
- Add table printout for the test client or present the result in a nice way (you don’t need to deploy the client
- Add a readme just with some sentences describing what you have done.
- Add the code both for the server and the client, also include a readme in the archive and upload everything to canvas.
    - Do not include” node_modules” directory.

If you don’t like JavaScript you can rewrite the client and the server in some other language,

EXTRA CHALLENGE (NOT MANDATORY)

Deploy the same backend several times on Heroku as different apps. Modify the client so that it can send requests targeted several instances of your backend.

Use a textfile with some more words in it and divide the data into several chunks and then send to different instance of your server.

Combine the results from the different backends into one result in the end at the “client”

You could also add an authentication procedure, so that the client needs to authenticate against the server before any data can be processed.