

You are tasked with creating a simple **single-page application** following the requirements handed to you. This task is not meant to be strict, if you find yourself having to make a decision, just choose whatever you think benefits you the most.

## Requirements

Given a chat log file(.txt) and a number  $k$ , determine the top  $k$  *chattiest* users.

- Chattiest means how many words a user has chatted (typed in)
- A word is simply anything between whitespace.

The logfile format is like this:

```
<user1> this is some chat words
<user2> the sky is blue
This line is still attributed
This line is still attributed!
<user1> more chat from me!
<user3> this is some chat words 1
<user3> this is some chat words 2
<user3> this is some chat words 3
<user2> the sky is blue
This line is still attributed
<user1> more chat from me!
This line is still attributed!
```

You can assume the characters “<” and “>” only appear in the file for attributing a username.

**Create a web/mobile app which has:**

- A one page front-end to upload a .txt file. You’re free to use any library or framework but Next Js is recommended. It should look roughly like this:

The wireframe shows a rectangular container with a black border. Inside, on the left, is a section titled "Upload a log file (.txt)". Below the title is a dashed rectangular box containing the text "(Drag and drop your file here)". Below this box is a rounded rectangular button labeled "Upload". To the right of the upload section is a section titled "Results:". Below this title is a rectangular box with an orange border containing a list of results:

- 1. User1 - 100 words
- 2. User2 - 90 words
- 3. User3 - 80 words
- User4 - 80 words
- 4. User5 - 70 words

- Return a list of the **chattiest** users, in **descending** order
- **Validate the file type** of a file upload, prompt error to user if it's not a txt.

**Optional task: *[completing this will get special considerations]*:**

- You separate your application into 2 independent parts, one for frontend and one for backend.
- Your application allow multiple file upload, and display results accordingly
- You deploy your application to cloud server (any cloud server)
- You write tests for your application.

## What we expect from you

- You write efficient code, reusable code and efficient loop
- You handle errors and exceptions
- Your application is well structured

Please include all applicable tests, push your code to GitHub or any Git repository hosting service and **send us the link**. As a final note, remember that this is more about **showcasing your knowledge** than about **solving a problem**.