

Problem Statement:

Yenmo wants to create a web application that displays live logs from a `.log` file (formatted as `.txt`). Currently, developers use the `tail -f website.log` command in the terminal to achieve this functionality. This command shows the last 10 lines of the `website.log` file and continuously listens for any changes in the file. Whenever a new line is added to the log file, it is immediately displayed in the terminal. However, this method is not user-friendly for non-technical business personnel. The developer has been assigned the task of creating a web interface that provides the same functionality as `tail -f`, but without requiring any coding knowledge.

Input:

sample.log file (attached).

Expected Output:

A folder to be shared on Google Drive containing the `website.log` file and the web application files. When the web application is run, it should be accessible on `localhost:3000` and display the last 10 lines of the `website.log` file. Furthermore, it should dynamically update the displayed logs to show any new changes in real-time.

For the Candidate:

- The candidate has a time limit of 3 hours to work on this problem statement.
- Building the entire application within this timeframe is not expected. The main objective is to assess the candidate's resourcefulness and prioritization skills.
- If possible, the candidate can host the application on free hosting platforms.
- The final 15 minutes will be allocated for discussing the candidate's thought process and exploring potential improvements for the application.
- The candidate is allowed and encouraged to use the internet, including resources like GitHub and Stack Overflow, to aid their development process.
- While the use of AI tools such as ChatGPT is not preferred, the candidate can utilize them if they encounter any difficulties. However, the final product and its quality are the primary focus.

Suggestions (from the company, but the candidate can deviate from these):

- Use React or Next.js to build the frontend.
- Utilize WebSockets to listen for changes in the log file.
- The backend can be implemented in Java, Node.js, Python, or any other language that the candidate is comfortable with.

Best of luck! This real-world use case will provide you with a glimpse into the tasks you will handle independently if you join our company. Enjoy the challenge, and feel free to reach out if you have any questions or require assistance with the problem statement.