Technical Assessment

Brief

Welcome to the take-home assignment for the role of a **Full Stack Product Engineer**. Your task is to build a reactions feature.

This is a full-stack technical assessment. We are assessing how you work on the frontend, backend and devops for building this application out.

Difficulty: Medium
Duration: 4 - 6 hours

Technical Requirements

1. React 18 + Next.js 14 (app directory)
2. Pick either Prisma or Drizzle for your ORM
3. Use either Supabase or Neon for your database
4. Use TailwindCSS and Radix for your components
5. Typescript (Strict)

Important Notes

1. Please do not share this assessment with anyone.
2. This assessment seems easy but we're looking closely at how you pay attention to the details.

3. The designs are an example of what is obtained from a client. It may be incomplete, and it's sometimes up to you to fill in the blanks.

Evaluation Criteria

Your submission will be evaluated based on the following criteria:

- **Correctness**: Does your application fulfill the specified requirements?
- **Code Quality:** Is your code well-organized, readable, and maintainable? Is the component reusable – how extensible is the component API?
- **Architecture:** Is your architecture design simple, intuitive but adaptable for future needs?
- **Documentation**: Are your instructions clear, and is the code accompanied by proper documentation?
- **Developer Hygiene:** Are commit messages semantic? Is the repo organized in a way where other devs can easily work on it?
- **Bonus Points**: Any additional enhancements or considerations will be considered for bonus points.

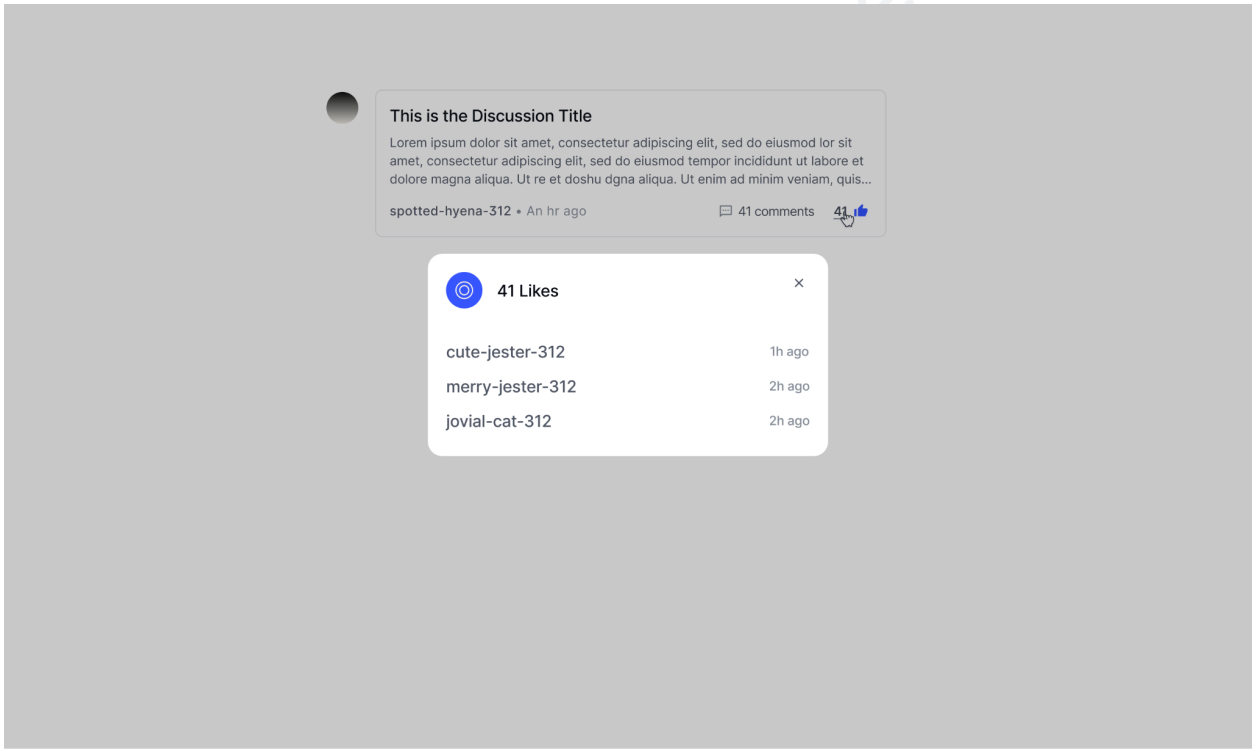
Deliverables

- **Submission Deadline:** please submit your completed assignment within 1-3 days of receiving it. If you have any questions or need clarification, feel free to reach out.
- A **full-stack application** in the technical stack specified
    - Make sure the components have an **intuitive component API**
    - **Reactions must be extensible**. It can be used for other models such as Comments, Activity, etc.
    - **Attention to details and UX** – does it look exactly like the Figma? Bonus points if you improved the UX!
- A **video** of the working demo walking us through how you approached the project – of the application and the codebase. We recommend using Loom.
    - Keep it **under 5 mins**. Do not add background music, or post-edits. Just record it and explain how it works, with your thought process behind how it was designed.
- A ***private*** **repository** on Github for us to clone the project to assess your code
    - Add the following users to your repo
        - https://github.com/gczh
        - https://github.com/Michaelomh

Reusable Reactions Component

Different reaction states

List of users who reacted



Here's the [link to the Figma file](#).

Bonus

Please do not attempt this unless you have completed all the core requirements. The bonus section is for the courageous who want some challenge. If you do not attempt it, you **won't be penalized**.

- Use hono.js for hosting the API (secured)
- Implement testing with modern testing frameworks (Jest, RTL, Vitest)
- Infinite scroll / pagination for reactions list
- Storybook for your components

Best of luck! We look forward to reviewing your work.