

Frontend Coding Exercise

Hi 🙌 Welcome to H2's frontend coding exercise. This is part of our hiring process for frontend engineers. We appreciate you taking the time to participate, and we ensure your submission will be carefully reviewed by humans with equal attention and effort.

Before we begin

This exercise reflects our engineers' actual work at H2. Unlike traditional, arbitrary algorithm questions that could be done easily by AI, this exercise focuses on decision-making, product sense, and overall engineering skills that could not (yet) be replicated by AI.

This means a few things:

1. **There will be no detailed clarification.** How you fill the gaps is the core of the exercise. We do provide demos (video, even!) for better communication, but they should be treated as competitors' references, not our own design.
2. **You can use AI.** If you chose to do so, it should go without saying that you are responsible for all of your AI output, including any issues or inconsistencies that could impact your interview result.
3. **Your communication is as important as your code.** Your submission should reflect not only your best code, but also why you decided to do so. For each task there should be thousands of ways to execute, and we want to know how you choose one.

This exercise is designed to be done in 2 days of full-time work. However, as we see that most of our candidates are high-performers at their current job during the day, it's okay to carry this out in a series of evenings as well. In all cases, do let us know your estimation after reading the exercise.

Your tasks

We will work on a first version of an imaginary web app. This app is built around a canvas that is powered by tldraw. Below are:

- What we need from the app
- Why we want you to work on it, and
- What we want to see from your work.

There are 5 tasks.

Task 1

First, we need a new repository. Even though we may not need to initialise a new project every day, the setup of a repository impacts every single day after. We want to see your opinions on package management, code formatting, and folder structure.

Task 2

Second, we need to display a PDF. PDF rendering is not only our very first impression, but a core one throughout the app, as all user interactions happen on top of it. We want to see how you make tradeoffs and optimisations, especially in terms of performance.

Task 3

Third, we need a basic tool. Custom shapes and tools are the heart of both our frontend engineering and tldraw itself. We want to see how you could make use of extensive documentation.

The tool should create a shape that:

1. Has a button whose label should display the shape size in real time.
2. Has a label that displays how many times the button is clicked.

Task 4

Fourth, we need a pin tool. Real-life products often require deep integration with the Editor API from tldraw. We want to see how you can utilise not only the latest tools but also their latest functionalities.

The tool should create a shape that:

1. Looks like a pin.
2. If the pin is on top of 2 overlapping shapes, the 2 shapes are "attached". That is, moving one will move the other as well.

Task 5

Finally, we need a camera tool. Our products don't stop until the user has their data back to them, in formats that are familiar and universal. We want to see how you integrate different APIs to produce a seamless user experience.

The tool should create a shape that:

1. Lets the user "crop" a part of the canvas,
2. Then export that part as an image.

That's it!

At this point, you should have a good understanding of the work, and thus a rough estimation based on your current experience. As mentioned, do let us know if you prefer to allocate 2 – 3 focused days to work on it, or to carry this out throughout the week.

To submit your work:

1. Exclude files that should be ignored in git,
2. Include anything that should be beneficial for your communication,
3. Zip everything and send it as a response to our email.