Take home test

This project serves as the starting point for the 'take home' evaluation. You will have a week to complete this project. The due date isn't completely set in stone. If you need another day or two that's not an issue. If you have any questions feel free to reach out.

Getting started

You will be updating this project to complete two 'features'. The main files you need to edit are `pages/index.tsx`, `pages/api/post.ts`, `pages/graph.tsx` and `pages/notes.tsx`. Feel free to change other files though. After you are done zip the project and send it back to us.

Goals

Page: Graph

Create a page that displays the provided `testData` in a line graph. The line graph's x-axis and y-axis should be `timestamp` by `fps` respectively. Each distinct `user` should get its own line. Hovering a data point should reveal a tooltip that displays all of that data-point's properties.

Page: Notes

Create a page that allows users to read, delete, and create 'posts'. Posts should be made of two properties, `author` and `body`. These posts should be saved in a database. Display these notes in a 'feed'. Think Facebook or Twitter. Don't worry about user login or anything like that. Author can just be a free form field.

The API should follow standard `REST` practices.

General: Design

The site should feel a bit more 'completed' than just raw html forms. Clearly the content of this site wont look like a cohesive product, but you should make an effort to give the site a proper layout.

Deployment

Host your completed work somewhere that you can link to.

Things you will be evaluated on.

- General code quality (should be easy to read)
- Completeness of page features. Complete as much as you can.
- Consider edge cases. For example, submitting an empty form shouldn't be allowed.
- User experience. You don't need to be a UX/Design pro, but the site should be trivial to use, and feel like a modern-ish app.

Suggestions

Using these libraries/tools doesn't give you any extra points. Feel free to use what's comfortable to you. This list is provided to help you out if you aren't sure where to start.

- Vercel (site hosting)
- Prisma (ORM for database connections. Prisma Cloud for database hosting.)
- MUI (design system)
- react-hook-form (form state management)
- zod (schema validation)
- SWR (promise state management)
- chart.js / react-chartjs-2 (chart drawing)