Full stack developer - Test assignment

Thank you for your interest in the Full Stack Developer position with the Lumion Cloud team.

To properly assess your technical skills and provide a relevant context for a follow-up conversation, we ask that you complete a small test assignment.

Once you have reviewed the assignment details, please contact us to arrange a review and subsequent discussion.

Task

Please make a website that enables a user to post a message to a page e.g. a message board.

Requirements

Core

- **Messaging System:** Users must be able to post a new message and post a reply to an existing message.
- **User Session Management:** The application requires a session mechanism to distinguish between two separate users. Authentication is optional, and if implemented then we suggest using NextAuth.
- **Real-time Updates:** When a user posts a reply, all other users currently viewing that message thread must see the new reply immediately, without needing to refresh the page (i.e., using WebSockets or similar real-time technology).

Technical Stack

- **Frontend:** The implementation must be made in TypeScript and using React function components and hooks.
- **Data Fetching:** All data retrieval must be accomplished using GraphQL queries.
- **Data Persistence:** Data must be stored persistently. Acceptable options are: a local JSON file, an SQLite database, or a PostgreSQL database.

Delivery and Setup

- **Execution:** A guide on how to run the project(s) must be provided in a README file. The project should be runnable locally without any complex setup (e.g., third party account creation). The suggested run command is a simple terminal instruction like "npm run dev".
- **Submission:** The completed code must be delivered via a GitHub-hosted repository.

Suggestions

Although not a strict requirement, we recommend implementing the data structure and GraphQL interface using KeystoneJS, as this is the technology we currently utilize. Examples of GraphQL subscriptions relevant to this assignment are available on the KeystoneJS GitHub repository as well as numerous other examples.

For the sake of simplicity, you do not need to create a single monolithic project. Feel free to build a frontend project using technologies you are already familiar with and a separate backend project using KeystoneJS. This approach minimizes the learning curve for KeystoneJS if you are not already familiar with it.