Thank you for taking the time to engage with us. As discussed, this is the assessment for the **Senior Front-end Engineer** role. You have 7 days from the day you decide to commence.

Task Name

Real-Time Animation Editor with Collaboration

Objective

Your task is to create a mini real-time Lottie animation editor. The application should allow users to upload a JSON file and manipulate properties like speed, color, or scale. Additionally add support for layer management with capability to view and select layers and remove multiple layers. Moreover, multiple users should be able to edit the same Lottie animation simultaneously in real-time.

The application should also support users to import an animation from the featured animations on LottieFiles website, utilize LottieFiles GraphQL API to fetch and showcase the featured animations for importing.

Technology Stack

- **React** as the frontend library
- **TypeScript** for type safety
- **WebSockets or similar server-push technologies** for real-time collaboration
- **GraphQL** for fetching featured animations
- **State Management Library** (e.g., Redux, MobX, Recoil)
- **Node.js with Express or Fastify** for back-end (if needed)
- **Any Database** for storing animation JSON and user collaboration state

Areas of Focus

React Proficiency: Structure your components efficiently. Use hooks or any modern React features that you find appropriate.

Real-Time Data Manipulation: Implement real-time data editing capabilities for properties like speed, color, and scale.

State Management: Choose an appropriate state management library and demonstrate how you would manage state across multiple components.

API Interactions: Fetch animations using either your own API or the LottieFiles GraphQL API. You may use the query field **featuredPublicAnimations** if you opt for the LottieFiles API.

WebSockets: Use WebSockets or any server-push technology to facilitate real-time collaboration between multiple users.

State Synchronization and Conflict Resolution: Implement mechanisms to synchronize state between users and resolve any editing conflicts that might arise.

Performance: Given that this is real-time collaboration, performance should be optimized. Demonstrate how you would approach this.

UI/UX: Even though design is not the primary focus, usability should be considered. If you wish to showcase your front-end skills, feel free to dazzle us.

Bonus

Implement a chat feature where users can discuss changes in real-time within the application.

Additional Notes

- Create a Git repository and document your project as you would for any professional endeavor.
- Although you don't need to set up any authentication, treat this as a real project.
- It would be beneficial to have a deployed version so we can evaluate the application interactively.

Resources

- https://lottiefiles.github.io/lottie-docs/ - Lottie format documentation
- https://lottiefiles.github.io/lottie-docs/playground/json_editor/ - Lottie JSON Viewer / Playground.
- https://graphql.lottiefiles.com - LottieFiles GraphQL API (Altair playground available at https://graphql.lottiefiles.com/graphiql)

Once you've completed the assessment, please provide us with access to your Git repository for evaluation.

Duration

You have 7 days to complete this assessment.

Feel free to ask any questions you may have.

Good Luck!