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

Task Name:

Offline-First Lottie Animation Management System

Objective

Your task is to build a small React application and required APIs that enables users to search and preview, upload and download Lottie Animations. The application must have a detailed animation view showing the basic metadata of the animation. The application must provide robust offline capabilities, allowing users to interact with animations and access detailed metadata even when no internet connection is available.

Technology Stack

- **GraphQL for API development** Develop a GraphQL API to handle search queries, file upload / download and metadata retrieval.
- **React** for the front-end
- **TypeScript** for static typing
- **Service Workers** for offline support
- **State Management Library** (e.g., Redux, MobX, Recoil)
- **IndexedDB or LocalStorage** for local caching
- **Node.js with Express or Fastify** for back-end (if needed)
- **Any Database** for storing animation JSON and user collaboration state (if needed)

Additional Functionalities

- **GraphQL APIt** Design and implement a GraphQL schema and resolvers for the following:
    - **Search / Browse animations** Allow users to query animations based on different criteria
    - **File Upload** Enable users to upload Lottie animations files from their computers

- **File Download** Allow users to download animation files to their computer

Areas of Focus

Service Workers and Caching: Implement service workers to provide offline support. Cache animations and other assets so they can be viewed offline.

Progressive Web App (PWA) Features: Make the application installable with a manifest and meet other PWA criteria.

GraphQL API Interactions: Implement the necessary front-end operations to interact with your GraphQL API.

State Management: Demonstrate how you would manage application state, especially with offline data and online data synchronization.

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.

React and TypeScript: Leverage modern React paradigms and TypeScript features for better code quality and maintainability.

Performance Optimizations: Ensure that the app is performant, particularly when switching between online and offline modes.

Bonus

Implement syncing with the local library when the app goes back online. Resolve any conflicts or duplicates intelligently.

Additional Notes

- Create a Git repository for the project and document your design decisions, API schema, and any other pertinent information as you would in a real-world project.

- While you don't need to implement authentication, consider this a professional project. Code quality and project documentation will be evaluated.
- If possible, deploy a working version of the application to facilitate our review process.

Resources

- https://lottiefiles.github.io/lottie-docs/ - Lottie format documentation

Duration

You have 7 days to complete this assessment.

Feel free to reach out if you have any questions.

Good Luck!