Technical Test: React Developer with Next.js and Tailwind CSS

Task 1: Advanced Next.js Architecture (30 points)

- Implement a micro-frontends architecture using Module Federation.
- Integrate an external analytics module as a federated module.
- Secure routes using NextAuth.js for authentication.

Task 2: Advanced Component Development (35 points)

Animated UI Components (20 points)

- Develop a dynamic dashboard component with charts and graphs to visualize data.
- Use a data visualization library (e.g., D3.js, Chart.js) for rendering charts.
- Implement smooth transitions and interactions within the dashboard.

Custom Hook (15 points)

- Create a custom hook for handling complex state management scenarios (e.g., data fetching, caching, and real-time updates).
- Use the hook in at least two different components to showcase its versatility.

Task 3: Large Dataset Handling (25 points)

- Fetch and display a large dataset (e.g., 10,000 records) efficiently on the homepage.
- Implement virtualization or pagination to handle the large dataset without compromising performance.
- Optimize the data fetching mechanism for minimal server load.

Task 4: Performance Optimization and Server-Side Rendering (25 points)

- Optimize the application for mobile performance using Google Lighthouse metrics.
- Implement server-side rendering (SSR) for critical pages to improve the initial loading time.
- Ensure efficient resource loading using strategies like pre-fetching or pre-loading.

Task 5: Advanced Features (30 points)

Real-time Collaboration (15 points)

- Implement real-time collaborative editing for a shared document.
- Utilize a technology like Firebase Realtime Database or Socket.io for real-time updates.

Advanced Styling (15 points)

- Implement a theming system with multiple theme options.
- Integrate Tailwind CSS JIT for optimizing the styling workflow and reducing the final bundle size.

Task 6: Testing and Documentation (15 points)

- Write end-to-end tests using Cypress for critical user flows.
- Provide comprehensive documentation covering architecture, data models, and instructions for development and deployment.

Submission Guidelines:

Share the link to the Git repository containing your code.
Provide a detailed README with instructions, architecture overview, and any additional information.
Ensure that your application is deployed, and share the deployment link (e.g., Netlify, Vercel).