Full-Stack Interview Task: Comprehensive Job Board Application

Please Note: This document is for internal hiring purposes only and should not be shared externally.

This task evaluates your ability to design and implement a full-stack application. You'll build a job board from scratch, showcasing your skills in backend development, database design, frontend user experience, and secure authentication/authorisation.

Project Goal

Develop a functional job board application that caters to two main types of users:

1. **Job Seekers:** Individuals looking for jobs who can browse listings, apply for positions, and track the status of their applications.
2. **Employers:** Companies or individuals who post job openings, manage their listings, and review incoming applications.

Technical Requirements

- **Backend:** Node.js with TypeScript and the Express.js framework.
- **Database:** PostgreSQL.
- **Frontend:** React.js.

Core Application Functionality

The application should provide the following user experiences and capabilities:

1. Job Browse & Viewing

- **Public Job Listings:** Any user, even those not logged in, should be able to **browse and view a list of all available job postings**. Each listing should provide essential details like job title, company, and location.
- **Detailed Job View:** Users should be able to **select any job listing to view its complete details**, including a full description, requirements, and any other relevant information.

2. User Authentication & Authorization

- **User Accounts:** The system must support user registration and login.
- **Login Methods:** Users should be able to authenticate using **both** a traditional **email and password** (with secure password storage) and a **Single Sign-On (SSO)** method. You'll need to integrate with **one OAuth2 provider of your choice**.
    - **SSO User Management:** When a user logs in via SSO, your system should **create a new user account if one doesn't already exist**, or **link to an existing user account in your database** based on the SSO provider's unique ID.
- **Role-Based Access:** The application must differentiate between **"Job Seeker"** and **"Employer"** roles. User interface elements and backend actions should be restricted based on the logged-in user's role. For example, only Employers can see options to post new jobs.
- **Logout:** A clear mechanism for users to log out.

3. User Profile Management

- **Job Seeker Profile:** Logged-in Job Seekers should be able to **view their own profile information** and **edit and update their details** (e.g., contact information, basic resume summary).
- **Employer Profile:** Logged-in Employers should be able to **view their own profile information** and **update their details** (e.g., company name, contact person).
- **Viewing Other User's Info:** Employers should be able to **view relevant information about Job Seekers who have applied to their jobs** (e.g., contact details, uploaded resume/cover letter).

4. Employer Features

- **Job Posting Management:**
    - **Create Jobs:** Employers should be able to **post new job listings** through a dedicated interface, providing all necessary job details.
    - **Manage My Jobs:** Employers can **view a list of all the job postings they have created**. From this list, they should be able to **edit existing job details** or **delete** job postings that are no longer needed.
- **Application Review:**
    - Employers must be able to **access and review all applications submitted for their specific job postings**.
    - **Update Application Status:** Employers need the ability to **change the status of any application** they receive (e.g., mark it as "Under Review," "Interview Scheduled," "Rejected," or "Hired").

5. Job Seeker Features

- **Apply for Jobs:** Logged-in Job Seekers should be able to **submit applications for job postings**. This typically involves providing their contact information and uploading documents like a resume or cover letter.
- **Track My Applications:** Job Seekers must be able to **view a consolidated list of all the jobs they have applied for**. For each application, they should see the job title and its current status as updated by the employer.

Setup and Submission Guidelines

- **Code Repository:** Provide a link to a Git repository (e.g., GitHub, GitLab) containing your complete solution.
- **Local Development Instructions:** Your `README.md` file must contain clear, concise instructions on how to set up and run both the frontend and backend components of the application locally. This includes steps for setting up the PostgreSQL database (e.g., creating the database, running migrations).
- **Design Explanations:** In your `README.md`, briefly explain your key design decisions, including:
    - Your chosen database schema (tables, relationships).
    - Your overall authentication strategy (how password-based and SSO login work together, how user sessions are managed).
    - The OAuth2 provider you integrated and why.
    - Any significant architectural choices made in either the backend or frontend.