Project Overview

Build a Job Management Portal that enables companies to post jobs, applicants to apply with resumes, and recruiters/admins to manage applications via a dashboard. The system must be secure, responsive, scalable, and production-ready.

Core Features

1. **User Roles**
    - **Admin:** Manage companies, recruiters, and system settings.
    - **Recruiter:** Create job postings, view and manage applications.
    - **Applicant:** Browse jobs, apply, and track application status.
2. **Company Management**
    - Create and edit company profiles (name, address, website, description, logo).
3. **Job Postings**
    - CRUD for job postings (title, description, location, job type, salary, skills).
    - Public job listing and detailed job view.
    - Job search with filters (location, company, job type, salary, posting date).
4. **Application & Applicant Management**
    - Applicants apply with cover letter and resume upload (PDF/DOC).
    - Recruiters view/manage applications, update statuses (Received, In Review, Interview, Rejected, Hired).
    - Applicants can track their application status in a dashboard.
5. **Recruiter Dashboard**
    - Metrics: total jobs, open positions, applications received, hires made.
    - Application pipeline (drag-and-drop Kanban board).
    - Recent applicants and job-specific stats.
6. **Resume & File Handling**
    - Resume upload and secure storage (local or cloud).

7. **Notifications**
    - Email confirmation for applicants on submission.
    - Notification to recruiters when a new application arrives.
8. **Analytics & Reports**
    - Charts and tables showing job posting trends, applications per job, time-to-hire.
    - Exportable reports (CSV/PDF).
9. **Security & Compliance**
    - Authentication & authorization with JWT.
    - Password encryption (bcrypt).
    - Input validation and secure file handling.
    - Audit logs for job postings and application status changes.
10. **API**
    - REST APIs for all core modules.
    - Consistent error handling and response formats.
11. **Frontend (ReactJS)**
    - Responsive design (desktop, tablet, mobile).
    - React Router for navigation.
    - Forms with validation (React Hook Form).
    - Integration with backend APIs (Axios/fetch).
    - Components for tables, charts, modals, notifications.
12. **Backend (Java/Spring Boot)**
    - Core Java for models and business logic.
    - Spring Boot for API endpoints, services, and MVC.
    - JPA/Hibernate for database persistence.
    - Spring Security for authentication/authorization.
    - API endpoints with CRUD + search/filter.

13. **Database**
    - MySQL/PostgreSQL with normalized schema.
    - Tables: Users, Companies, Jobs, Applicants, Applications, Files, Notifications, Audit Logs.

Expected Deliverables

- Full source code (frontend + backend) in Git repository.
- Database schema and migration scripts.
- Responsive frontend integrated with backend APIs.