Backend Java Developer Assessment

Overview

You are tasked with creating a mini version of a recruitment marketplace system that connects organizations, jobs, and candidates. This system should allow organizations to:

- Create and manage jobs
- Assign candidates to jobs
- Establish vendor-client relationships with other organizations

Core Requirements

1. Organization Management

Provide RESTful APIs to manage organizations with the following attributes:

- ID (auto-generated)
- Company name
- Address
- City
- Logo (optional)
- Contact info (email, website)
- Account manager reference
- Creation timestamp
- Status (active/inactive)

2. Vendor-Client Relationship

Organizations can act as vendors and/or clients. APIs should support:

- Assigning a vendor to a client organization
- Listing all vendors for a specific client
- Listing all clients for a specific vendor

3. Job Management

Jobs are created on behalf of organizations with the following attributes:

- ID (auto-generated)
- Job title
- Description
- Required skills
- Experience level
- Organization ID (owner)
- Status (open/closed)
- Creation timestamp

4. Candidate Management

Candidate attributes:

- ID (auto-generated)
- First name
- Last name
- Email
- Phone number
- Resume/CV reference
- Primary skills
- Total experience
- Current location
- Preferred location
- Organization ID (owner)

5. Job Assignment

APIs to assign candidates to jobs with the following business rules:

- Clients can assign candidates from their vendors
- Track application status: APPLIED, SELECTED, REJECTED
- Validation to ensure valid assignments

6. Candidate Visibility

Organizations should see:

- Its own candidates
- Candidates from vendors assigned to its jobs

Technical Requirements

Backend Technology

- Java 11+
- Spring Boot
- RESTful API design
- JPA/Hibernate

Database

- MySQL or PostgreSQL
- Proper schema design with relationships
- Indexing for performance

API Documentation

- Swagger/OpenAPI

Authentication & Authorization

- Basic authentication

- Role-based access (ADMIN, ORG$_ADMIN,USER$)

Error Handling

- Exception handling with meaningful error messages
- Proper HTTP status codes

Bonus Features (Optional)

- Pagination & filtering on list APIs
- Candidate search by skills
- Reporting/metrics endpoints (e.g., job fill rate)
- Unit & integration tests
- Logging & monitoring

Submission Guidelines

- Create a GitHub repository for your project
- Add a README.md including:
  - Setup instructions
  - Swagger API docs or link
  - Database schema diagram
  - Design decisions explanation
- Ensure code is well-commented and follows Java standards
- Provide sample data/scripts for DB initialization

Evaluation Criteria

- Code quality & organization
- Database design
- API design & implementation
- Validation & error handling
- Security (auth & roles)
- Documentation quality
- Bonus features