**Job Portal Backend Application**

**Overview**

This application serves as the backend for a job portal. It includes user management, job postings, skill management, authentication, role-based access control, recommendation systems, and filtering systems.

**Table Structure**

**User Table**

- userid (UUID): Unique identifier for the user.

- name (String): Full name of the user.

- email (String, unique): User's email address.

- password (String, hashed): User's password.

- dob (Date): Date of birth of the user.

- address (String): Address of the user.

- education (JSON or separate fields):

- 10th: { yearPassed (Integer), percentage (Float), school (String) }

- 12th: { yearPassed (Integer), percentage (Float), school (String) }

- college: { yearPassed (Integer), percentage (Float), college (String) }

**Experiences Table**

- experienceid (UUID): Unique identifier for the experience record.

- userid (UUID): Foreign key to the User table.

- companyname (String): Name of the company.

- title (UUID, foreign key to Titles Table): Role held by the user.

- startyear (Year): Starting year of the experience.

- endyear (Year): Ending year of the experience.

- experience (Computed field): Calculated experience duration.

**Skills Table**

- skillid (UUID): Unique identifier for the skill.

- name (String, predefined values): Name of the skill.

**UserSkills Table (Many-to-Many Relationship)**

- userid (UUID): Foreign key to the User table.

- skillid (UUID): Foreign key to the Skills table.

**Job Posters Table**

- companyid (UUID): Unique identifier for the company.

- companyname (String): Name of the company.

- email (String, unique): Company's email address.

- password (String, hashed): Company's password.

- companylocation (String): Location of the company.

- companysize (Integer): Size of the company.

- companywebsite (String): Company's website URL.

**Jobs Table**

- jobid (UUID): Unique identifier for the job.

- companyid (UUID): Foreign key to the Job Posters table.

- jobtitle (UUID, foreign key to Titles Table): Title of the job.

- skillsrequired (Array or Many-to-Many relationship with Skills): Required skills for the job.

- experiencerequired (Integer): Required experience for the job.

- dateposted (Date): Date the job was posted.

- jobdescription (String): Description of the job.

- status (Boolean for active/inactive): Status of the job posting.

**Titles Table**

- titleid (UUID): Unique identifier for the title.

- title (String): Title name (predefined values).

**UserJobs Table (Many-to-Many Relationship)**

- userid (UUID): Foreign key to the User table.

- jobid (UUID): Foreign key to the Jobs table.

- status (String): Status of the user in relation to the job (e.g., applied, interviewed, hired).

**Features**

**Authentication**

- JWT Authentication: Secure API endpoints with JSON Web Tokens.

- Password Hashing: Hashing passwords for security.

**Role-Based Access Control**

- Define roles for users (e.g., user, company).

- Restrict access to certain API endpoints based on roles.

**Input Validation**

- Validate input data.

- Ensure email formats, password security, date formats, and other field-specific validations.

**Custom Exceptions**

- Implement custom exception handling for better error management.

**Logging**

- Use libraries for logging important events and errors.

**Recommendation System**

- Recommend jobs to users based on their skills.

- Match user skills with job requirements.

**Filtering System**

- For Users: Filter jobs based on skills, date, title, and company.

- For Companies: Filter applicants based on matched skill sets and experience in the applied title.

**Data Transfer Objects (DTOs)**

- Define DTOs for consistent data structure between client and server.

ER Diagram :

