CS3563: DBMS II
Group 9 Project Report

Darpan Gaur CO21BTECH11004 · **Abhinav Kompella** ES21BTECH11002 · **Aditya Bacharwar** ES21BTECH11003 · **Avaneesh Radhakrishnan** ES21BTECH11022

Introduction

The Job Portal system serves as a platform that bridges job seekers and employers. We have developed a web-based application for job seekers and employers to connect with each other. Job seekers can create profiles, browse job listings, apply for positions, and track the status of their applications. Employers can post job openings, review applications, and manage candidate information efficiently. The system also provides notifications to keep users informed about relevant events.

Database Design

ER Diagram: Figure 1 shows the ER Diagram of the database. The following tables are present in the database:

Functional Dependencies

- User_id $\implies$ Email, Password, Mobile [`Users Table`]
- Job_id $\implies$ Role, Description, Salary, Vacancy, Location, Industry, Company_id [`JobsListing Table`]
- Application_id $\implies$ User_id, Resume_id, Job_id, Status [`Applications Table`]
- Industry_id $\implies$ Industry_name [`Industry Table`]
- Company_id $\implies$ Company_name, Company_description [`Company Table`]

Technology Stack

- Frontend: HTML, CSS, JavaScript
- Backend: Django (Python)
- Database: PostgreSQL

Functionality

- User Registration/Sign-up/Login and Authentication
  - Implemented CustomUser Model extending AbstractBaseUser provided by Django, to allow users to login using email. Used Highly Secure Django's Authentication System for user authentication.
  - Extended User Sign-up to For Different Registration of Job Seeker and Job Provider.
  - This kind of Role-based Authentication allowed us to implement Role Based Access Control.
- Job Seeker
  - Job Seeker can create a profile with relevent details, upload resume, and apply for jobs.
  - Job Seeker can search for jobs based on location, Job Title, Company Name, Industry, etc.
  - Job Seeker can view and manage their job Application at one place. Allowing applicant Withdrew Application, View Application Status and other job relevent job details.
  - Job Seeker can view the list of jobs they have applied for.
- Job Recruiter
  - Job Recruiter can create company profile with relevent details, and post jobs.
  - Job Recruiter can view and manage their job postings at one place. Allowing recruiter to Edit Job Posting, Delete Job Posting, View Job Posting Status and other job relevent job details.
  - Job Recruiter can view the list of job seekers who have applied for their job postings.
- Notifications
  - User recieves notifications for various events like job posting, apllication status updates, etc.
  - Recruiters recieve notifications when a job seeker applies/withdraws from their job posting.
- Job Listing
  - All the jobs are listed on the home page.
  - Jobs can be filtered based on location, job title, company name, industry, etc.
  - Users can see the details of the job and apply for it.
- Resume Upload: Can upload resume in pdf format, and use that to apply for jobs directly.
- Additional tables: Support for additional tables like Industry, Location, Skill, etc.

Challenges Faced

- Implementing RBAC: To allow different users to access different parts of the application.
- Normalizing the database: To reduce redundancy and improve performance.
- Debugging: As lot of components were interacting with each other, it was difficult to debug the application.

Future Work

- AI based job recommendation system
- Resume Parsing and auto-filling application form

Conclusion

We have successfully implemented a Job Portal System that allows job seekers and employers to connect with each other. The system provides a user-friendly interface for users to interact with the application.

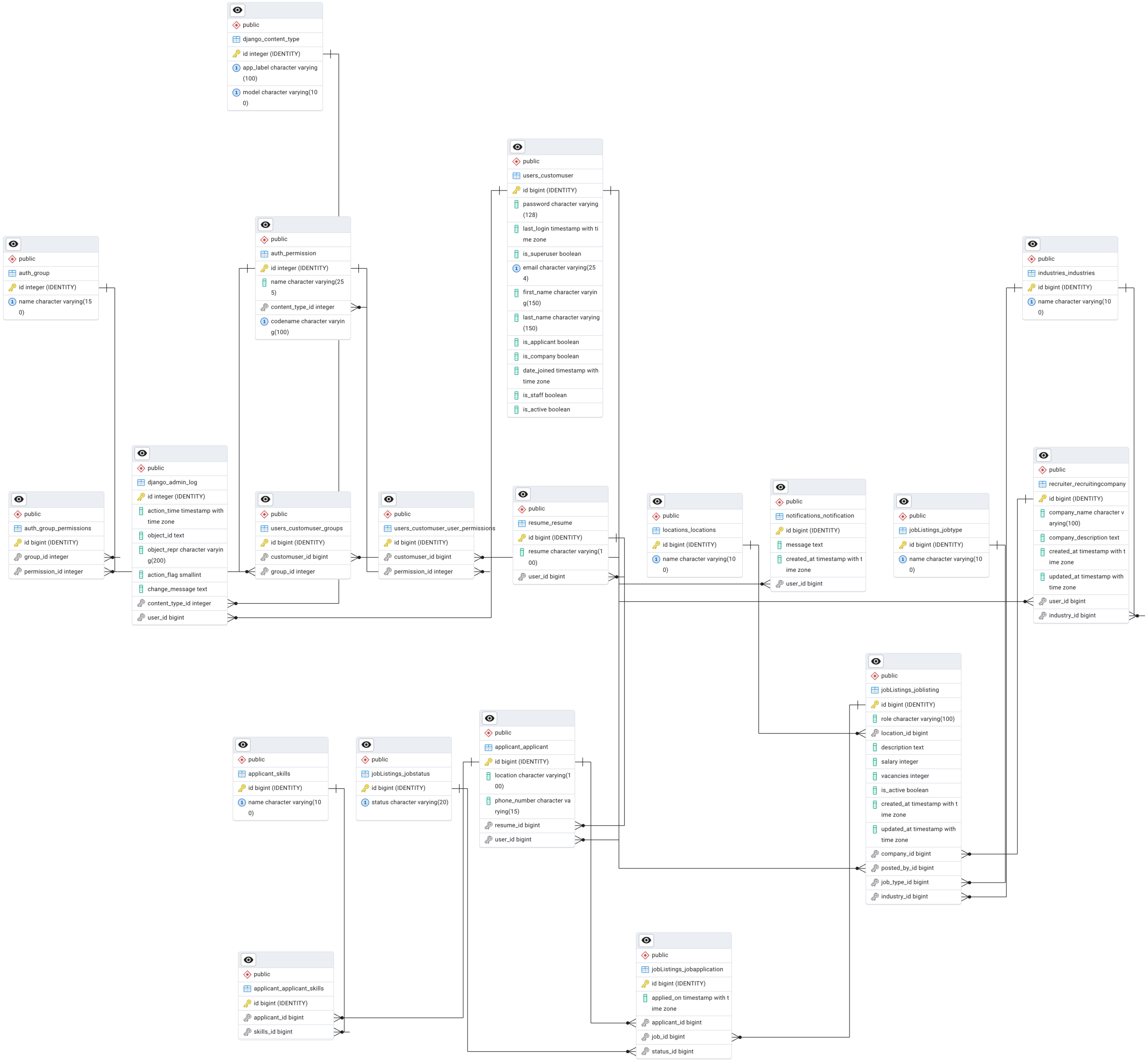

Figure 1: ER Diagram