**DOCUMENTATION**

By

Mahnoor Abbas (2312424)

Abdullah Waseem (2312102)

ACKNOWLEDGMENT

We are grateful that we were able to complete our project **Job Portal**, within the time allotted by our teacher, **Sir Abdil**. This achievement would not have been possible without the dedicated effort and collaboration of our group members, **Mahnoor Abbas** and **Abdullah Waseem**.

We extend our sincere thanks to **Sir Abdil** for his continuous support, guidance, and encouragement throughout the course, which played a vital role in the successful completion of this assignment.

Lastly, we would also like to express our heartfelt appreciation to our friends and all the respondents who supported us—either directly or indirectly—by taking the time to complete our questionnaires and provide valuable input.

**CONTENT**

**Introduction.............................................................................................................4**

**Objectives of the project.................................................................................................................... 5**

**Topic of Our Project.................................................................................................................... 6**

**Hardware/ Software Requirements..........................................................................................................7**

**WORK ANALYSIS......................................................................................................8**

**Database Schema....................................................................................................9**

**Project Snippets......................................................................................................15**

**ERD Diagram……......................................................................................................31**

***Introduction***

This project is a database-driven full-stack system designed to power a job portal platform. It efficiently handles users, job postings, applications, reviews, and associated skills. The design ensures scalability, data consistency, and clear relational logic between entities.

***Objectives of the Project***

The project is a job portal system that stores, manages, and connects information between job seekers, employers, and administrators through a relational database. It uses a set of interconnected tables to handle various operations like job posting, application tracking, user profiles, and skill matching.

The system includes tables for users, jobs, companies, applications, skills, categories, and reviews. It ensures structured data storage, efficient retrieval, and relationship handling between users and job-related entities.

Functions such as storing jobs, tracking application status, and allowing company reviews are managed through this database structure. The database is also capable of storing user types, controlling user activity, and supporting category and skill classification.

***Topic of Our Project***

The project is a Job Portal Database System designed to manage and organize job-related activities in a structured way. Database logic is required to handle entities such as users, companies, jobs, applications, and skills, as well as to control relationships between them. The system also includes user input through forms and queries, which allow for job postings, applications, skill tracking, and company reviews.

***Hardware/Software Requirements***

Since we developed the Job Portal Database using SQL, the essential piece of software is a database management system (DBMS), which might be one of several different programs like MySQL, PostgreSQL, SQLite, or Microsoft SQL Server. However, **PostgreSQL** is the DBMS that we are using. A laptop or PC compatible with the DBMS, in this case PostgreSQL, and sufficient storage space are the only requirements for the program's basic hardware.

***Work Analysis***

|  |  |  |
| --- | --- | --- |
| **Task** | Mahnoor Abbas | Abdullah Waseem |
| Analysis | ✅ | ✅ |
| Design | ✅ | ✅ |
| Coding | ✅ | ✅ |
| Testing | ✅ | ✅ |
| ­­­­Documentation | ✅ | ✅ |

### ***Database Schema***

#### 1. **Users Table**

**Description:**  
This code defines the main users table which stores account information for all users (job seekers, employers, and admins). Each user has a unique ID, email, password hash, user type, timestamps, and account status.

CREATE TABLE users (  
 user\_id SERIAL PRIMARY KEY,  
 email VARCHAR(255) UNIQUE NOT NULL,  
 password\_hash VARCHAR(255) NOT NULL,  
 user\_type VARCHAR(20) CHECK (user\_type IN ('job\_seeker', 'employer', 'admin')),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 is\_active BOOLEAN DEFAULT TRUE  
);

#### 2. **User Profiles Table**

**Description:**  
Holds additional personal and professional details about job seekers. This table is linked to the users table via a foreign key.

CREATE TABLE user\_profiles (  
 profile\_id SERIAL PRIMARY KEY,  
 user\_id INTEGER REFERENCES users(user\_id) ON DELETE CASCADE,  
 first\_name VARCHAR(100) NOT NULL,  
 last\_name VARCHAR(100) NOT NULL,  
 phone VARCHAR(20),  
 location VARCHAR(255),  
 resume\_url VARCHAR(500),  
 bio TEXT,  
 experience\_years INTEGER DEFAULT 0,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);

#### 3. **Companies Table**

**Description:**  
Stores information about companies created by employer users. Each company is linked to a user\_id who owns the company profile.

CREATE TABLE companies (  
 company\_id SERIAL PRIMARY KEY,  
 user\_id INTEGER REFERENCES users(user\_id) ON DELETE CASCADE,  
 company\_name VARCHAR(255) NOT NULL,  
 industry VARCHAR(100),  
 company\_size VARCHAR(50),  
 website VARCHAR(255),  
 description TEXT,  
 location VARCHAR(255),  
 logo\_url VARCHAR(500),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);

#### 4. **Categories Table**

**Description:**  
Defines job categories (e.g., IT, Finance, Marketing) to organize jobs and improve search filters.

CREATE TABLE categories (  
 category\_id SERIAL PRIMARY KEY,  
 category\_name VARCHAR(100) UNIQUE NOT NULL,  
 description TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);

#### 5. **Jobs Table**

**Description:**  
Contains job listings posted by companies. It references both companies and categories and includes job details such as salary, type, and status.

CREATE TABLE jobs (  
 job\_id SERIAL PRIMARY KEY,  
 company\_id INTEGER REFERENCES companies(company\_id) ON DELETE CASCADE,  
 category\_id INTEGER REFERENCES categories(category\_id),  
 title VARCHAR(255) NOT NULL,  
 description TEXT NOT NULL,  
 requirements TEXT,  
 salary\_min DECIMAL(10,2),  
 salary\_max DECIMAL(10,2),  
 job\_type VARCHAR(50) CHECK (job\_type IN ('full-time', 'part-time', 'contract', 'internship')),  
 location VARCHAR(255),  
 is\_remote BOOLEAN DEFAULT FALSE,  
 status VARCHAR(20) DEFAULT 'active' CHECK (status IN ('active', 'closed', 'draft')),  
 posted\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 deadline TIMESTAMP  
);

#### 6. **Skills Table**

**Description:**  
Stores skill names and their categories to help match users and jobs based on required expertise.

CREATE TABLE skills (  
 skill\_id SERIAL PRIMARY KEY,  
 skill\_name VARCHAR(100) UNIQUE NOT NULL,  
 category VARCHAR(100),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);

#### 7. **User Skills Table**

**Description:**  
Links each user to the skills they possess, including a proficiency level.

CREATE TABLE user\_skills (  
 user\_skill\_id SERIAL PRIMARY KEY,  
 user\_id INTEGER REFERENCES users(user\_id) ON DELETE CASCADE,  
 skill\_id INTEGER REFERENCES skills(skill\_id) ON DELETE CASCADE,  
 proficiency\_level VARCHAR(20) CHECK (proficiency\_level IN ('beginner', 'intermediate', 'advanced', 'expert')),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 UNIQUE(user\_id, skill\_id)  
);

#### 8. **Job Skills Table**

**Description:**  
Maps required skills to specific job listings. Each skill can be marked as required or optional.

CREATE TABLE job\_skills (  
 job\_skill\_id SERIAL PRIMARY KEY,  
 job\_id INTEGER REFERENCES jobs(job\_id) ON DELETE CASCADE,  
 skill\_id INTEGER REFERENCES skills(skill\_id) ON DELETE CASCADE,  
 is\_required BOOLEAN DEFAULT TRUE,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 UNIQUE(job\_id, skill\_id)  
);

#### 9. **Applications Table**

**Description:**  
Tracks all job applications made by users. It stores application status and timestamps for auditing.

CREATE TABLE applications (  
 application\_id SERIAL PRIMARY KEY,  
 job\_id INTEGER REFERENCES jobs(job\_id) ON DELETE CASCADE,  
 user\_id INTEGER REFERENCES users(user\_id) ON DELETE CASCADE,  
 status VARCHAR(20) DEFAULT 'pending' CHECK (status IN ('pending', 'reviewed', 'shortlisted', 'rejected', 'hired')),  
 cover\_letter TEXT,  
 applied\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 UNIQUE(job\_id, user\_id)  
);

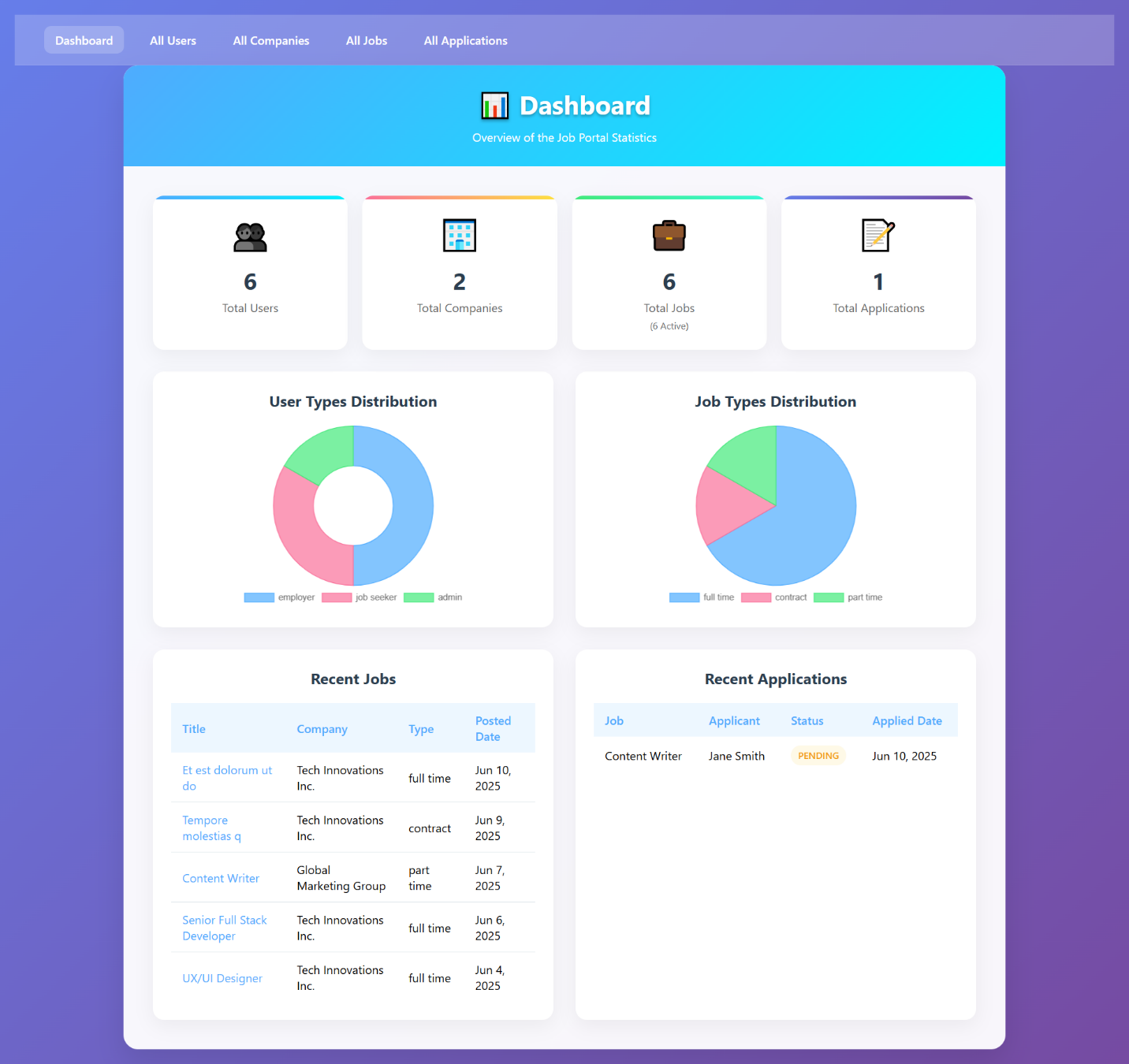
#### 10. **Company Reviews Table**

**Description:**  
Enables users to leave feedback and ratings for companies. Each review includes scores on work-life balance and salary benefits.

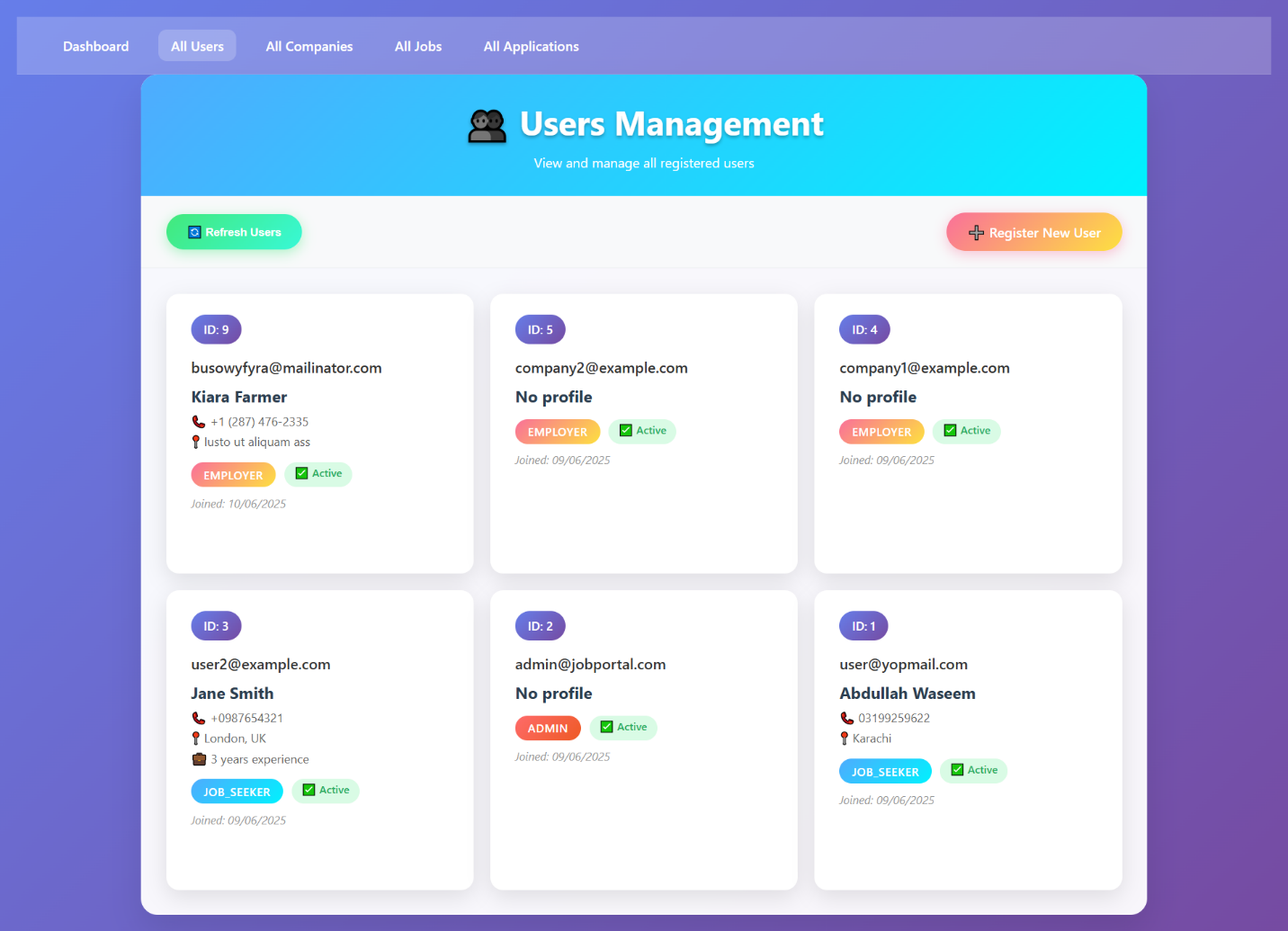
CREATE TABLE company\_reviews (  
 review\_id SERIAL PRIMARY KEY,  
 company\_id INTEGER REFERENCES companies(company\_id) ON DELETE CASCADE,  
 user\_id INTEGER REFERENCES users(user\_id) ON DELETE CASCADE,  
 rating INTEGER CHECK (rating >= 1 AND rating <= 5),  
 review\_text TEXT,  
 work\_life\_balance INTEGER CHECK (work\_life\_balance >= 1 AND work\_life\_balance <= 5),  
 salary\_benefits INTEGER CHECK (salary\_benefits >= 1 AND salary\_benefits <= 5),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 UNIQUE(company\_id, user\_id)  
);

### ***Project Snippets***

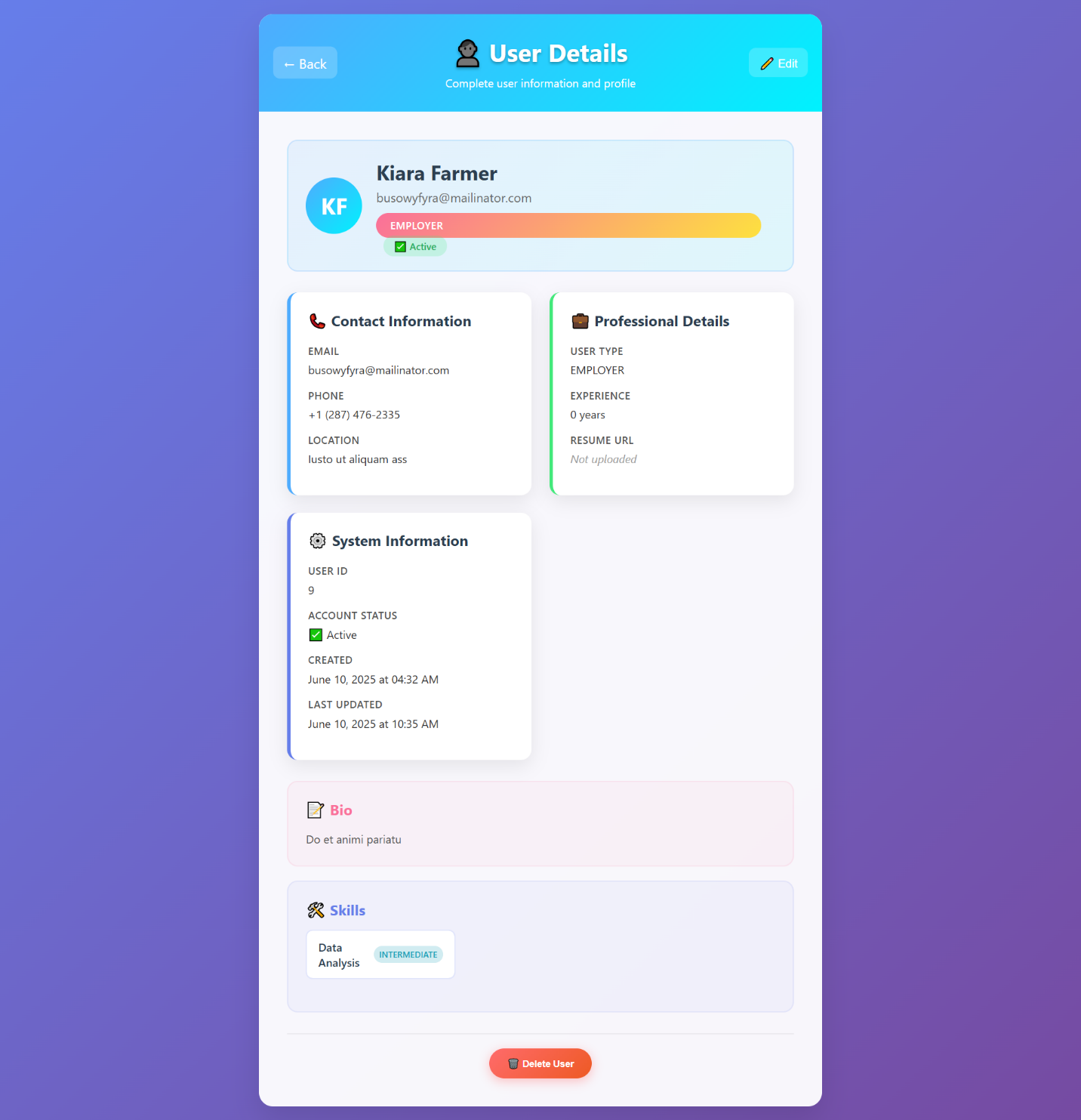
**Admin Dashboard:**



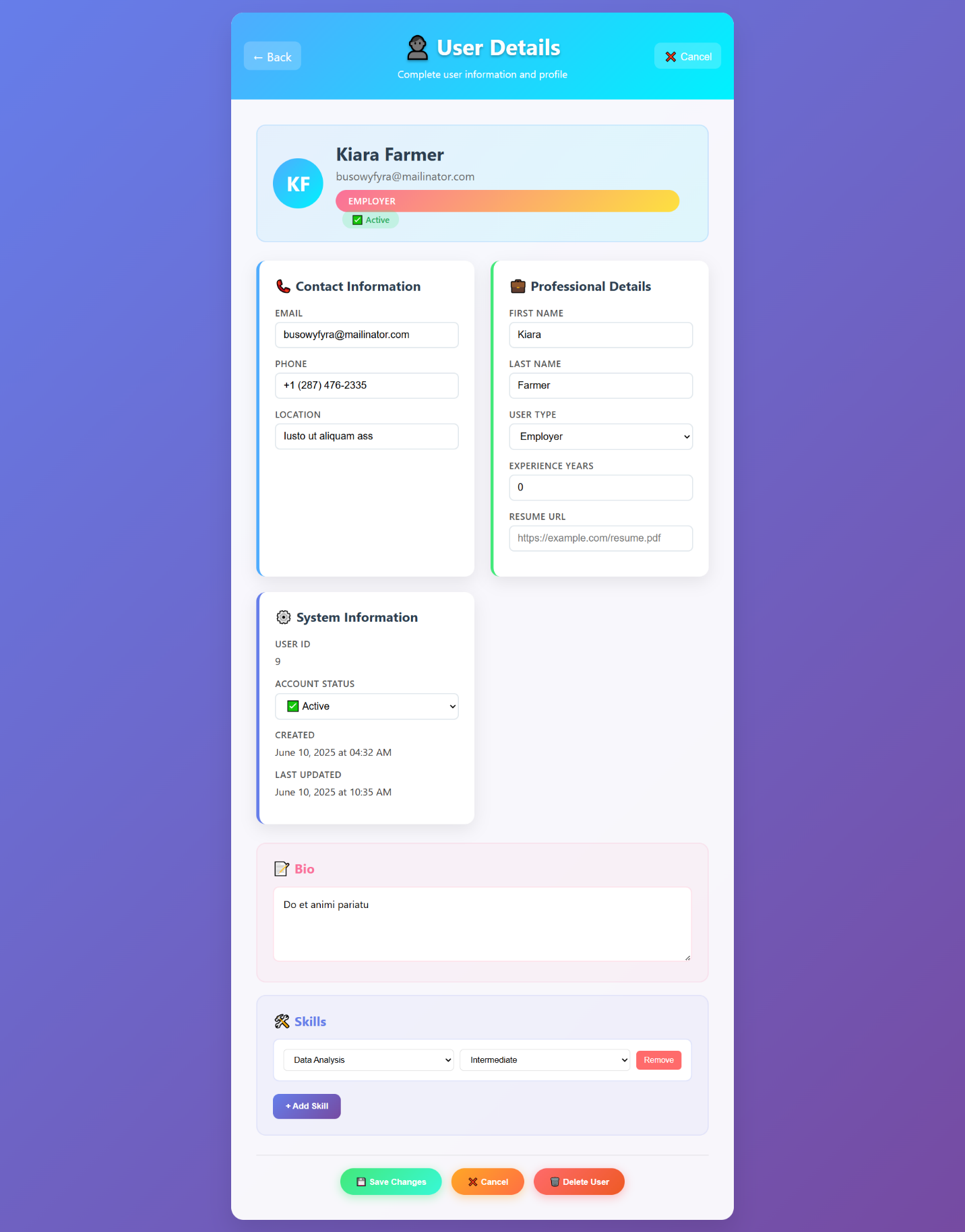
**All Users:**



**User Detail Page:**

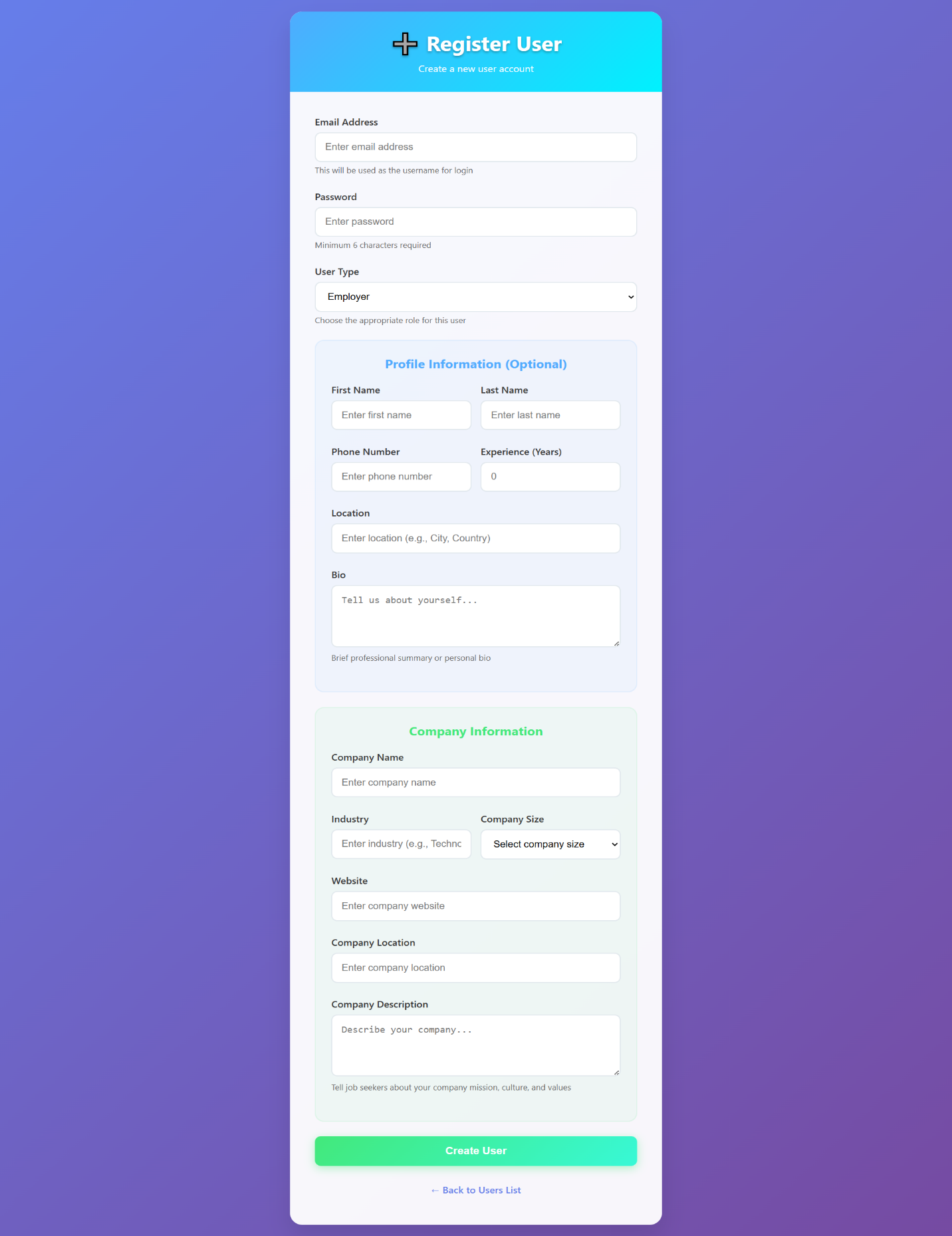


**User Detail Edit Page:**

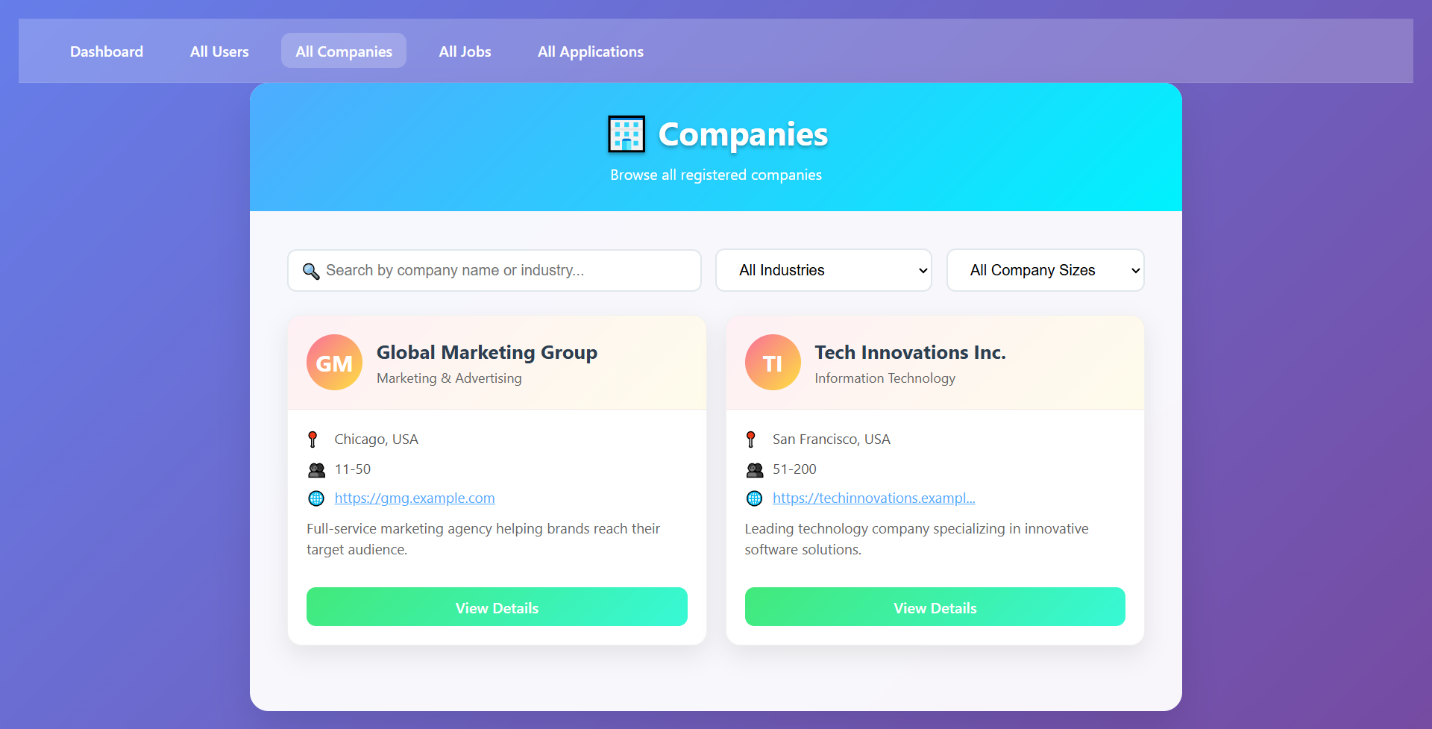


**Register User:**

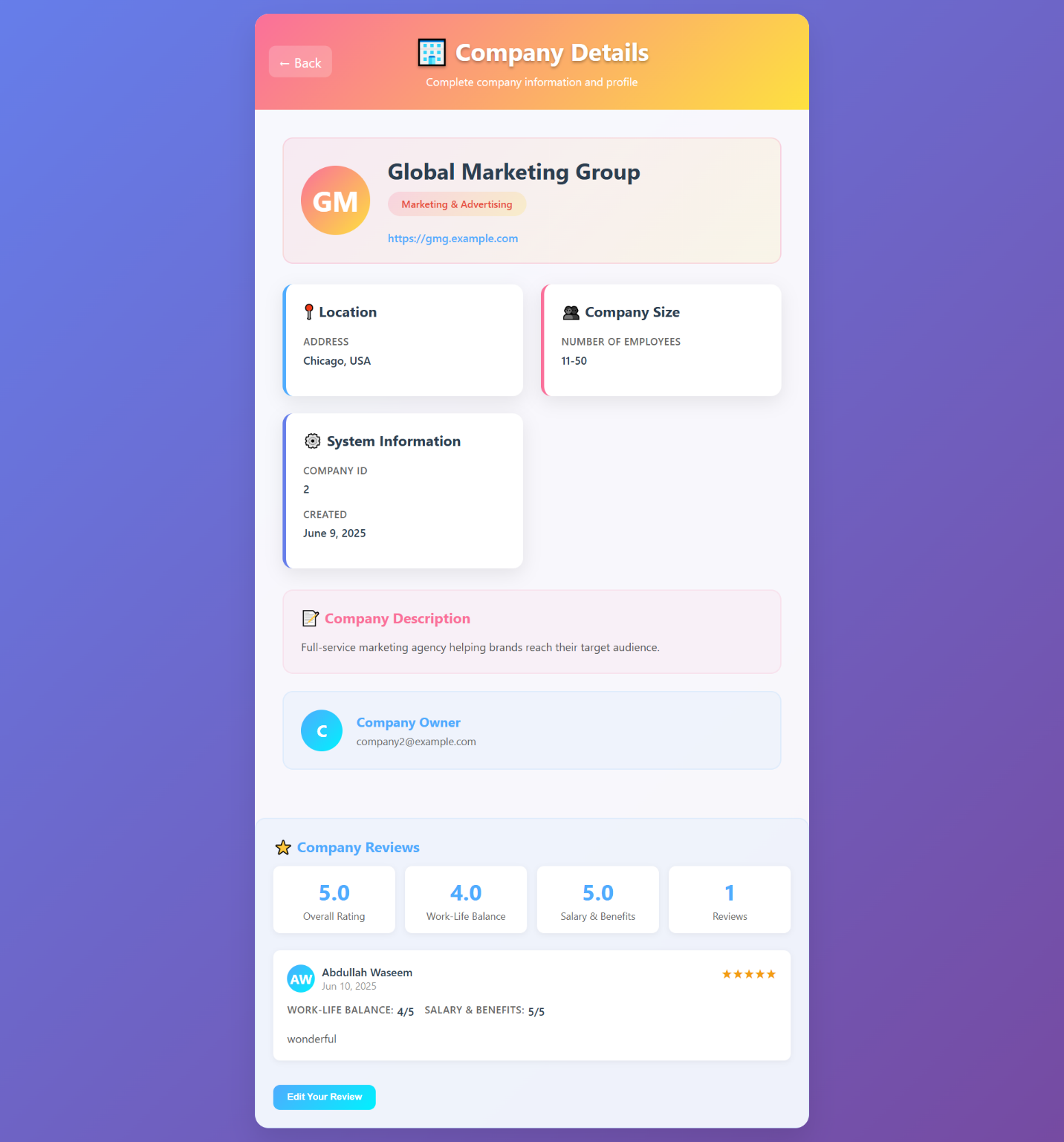


**Register Company:**

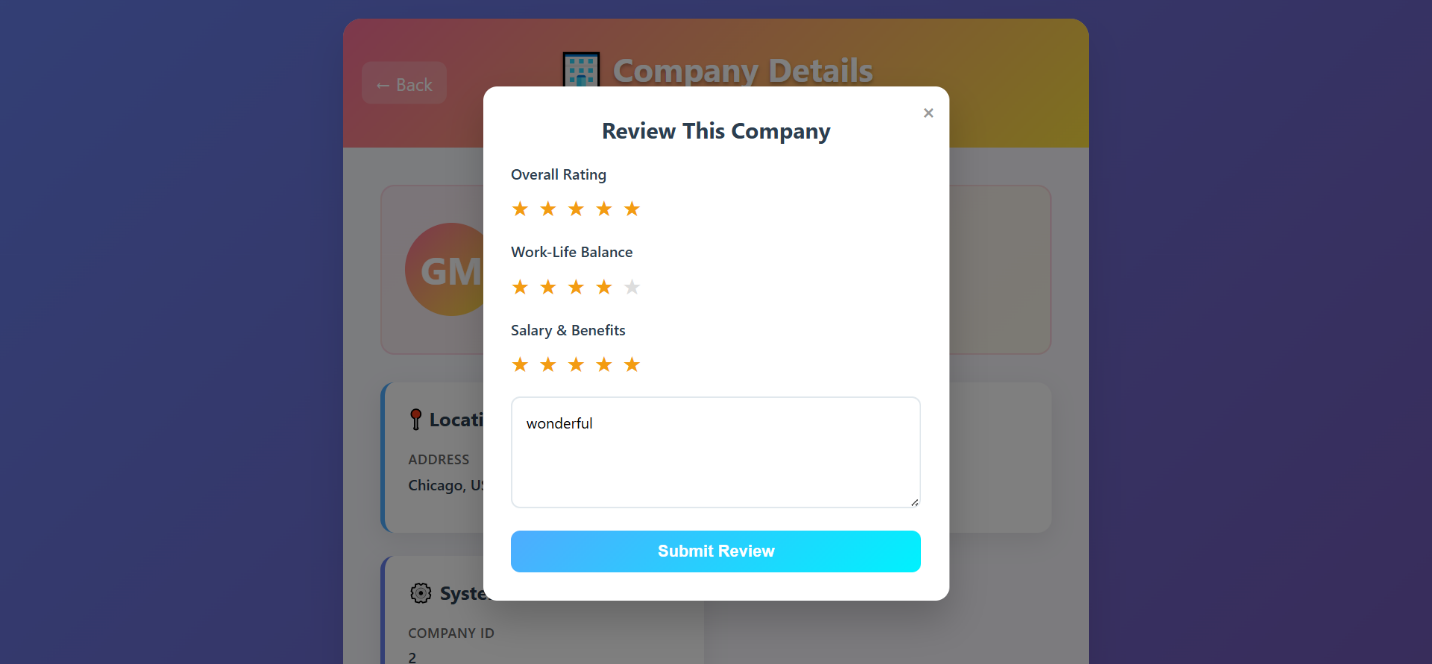
**All Companies**



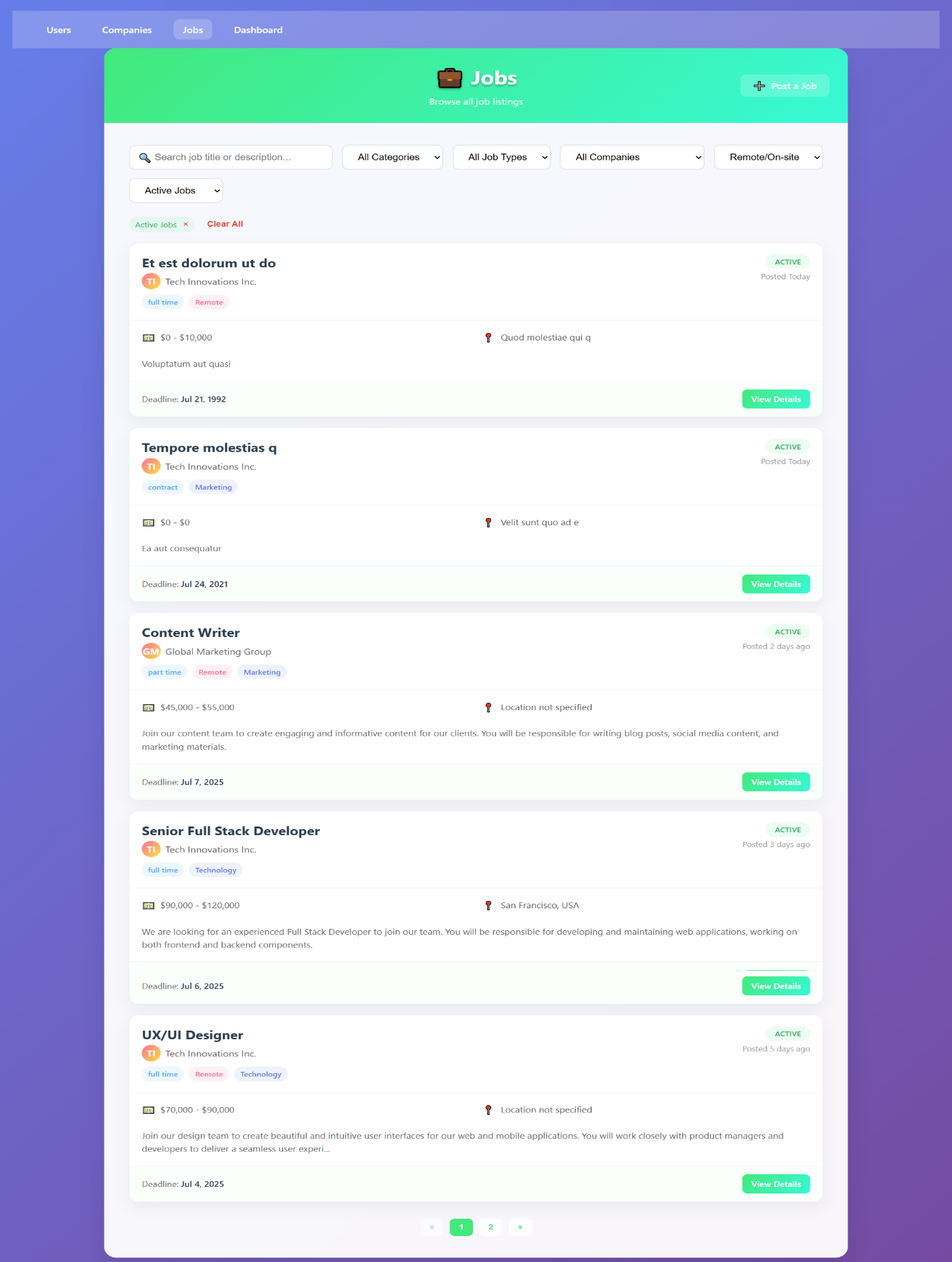
**Company Details and Review Section:**



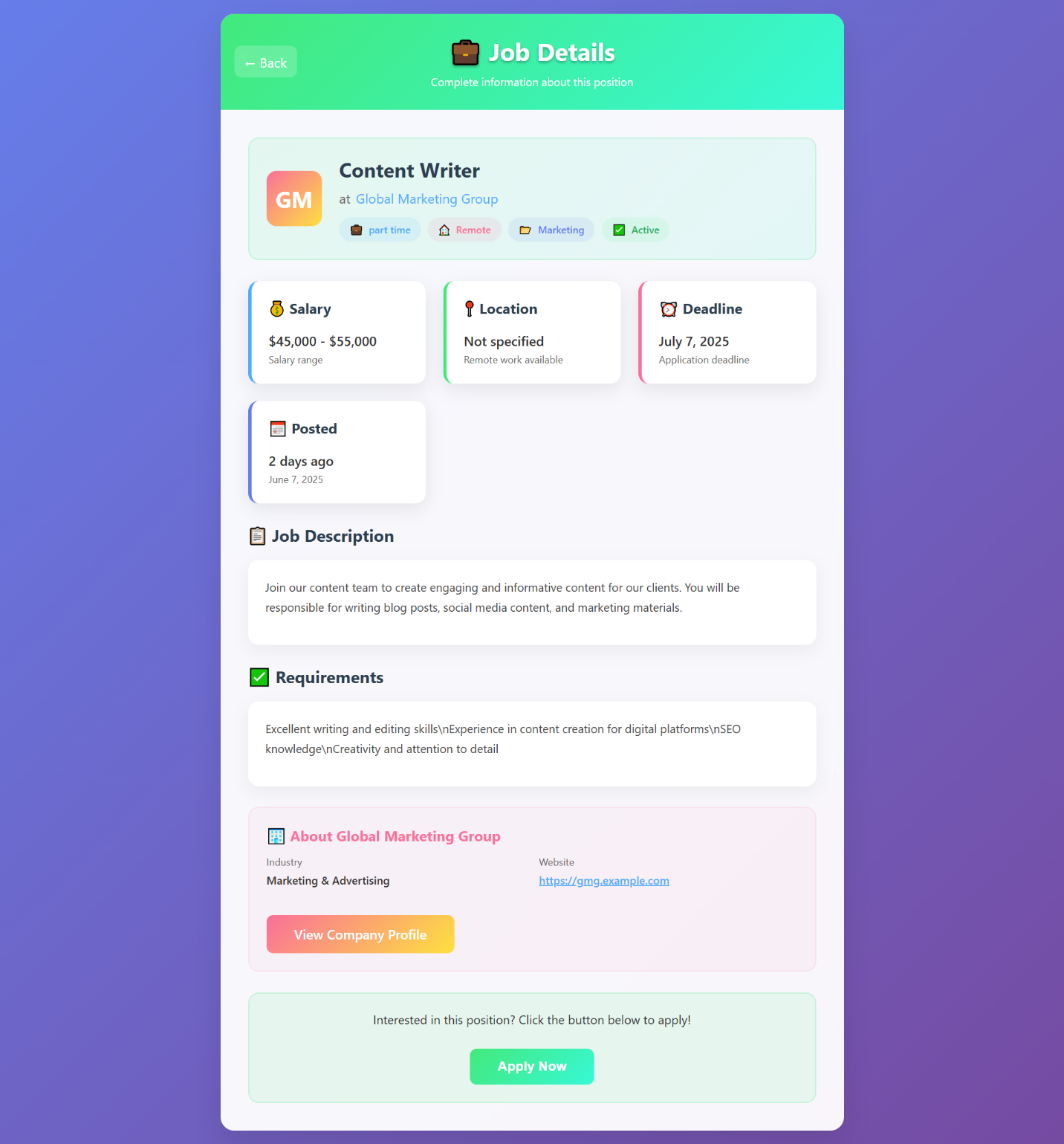
**Edit Review:**



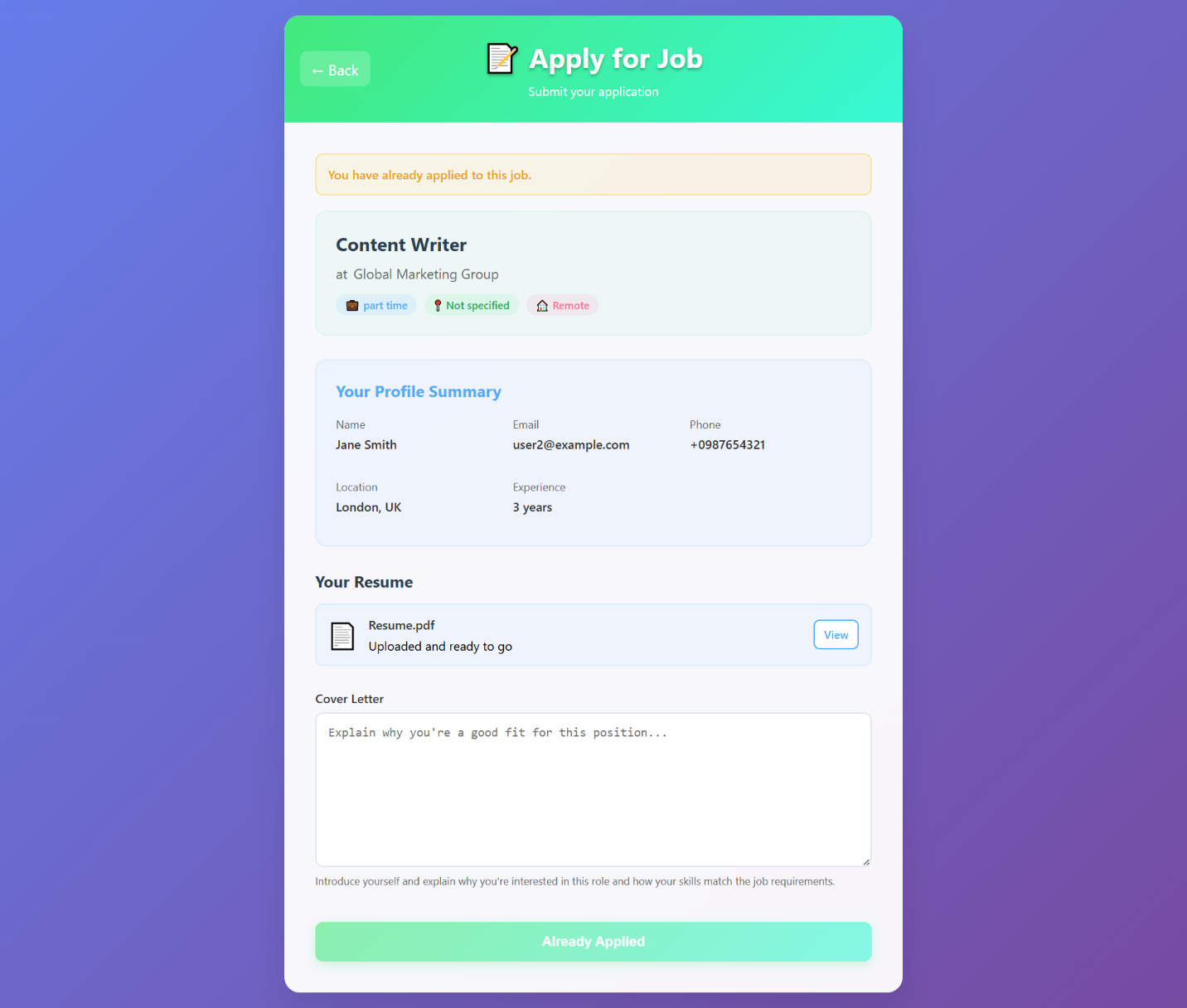
**All Jobs:**

****

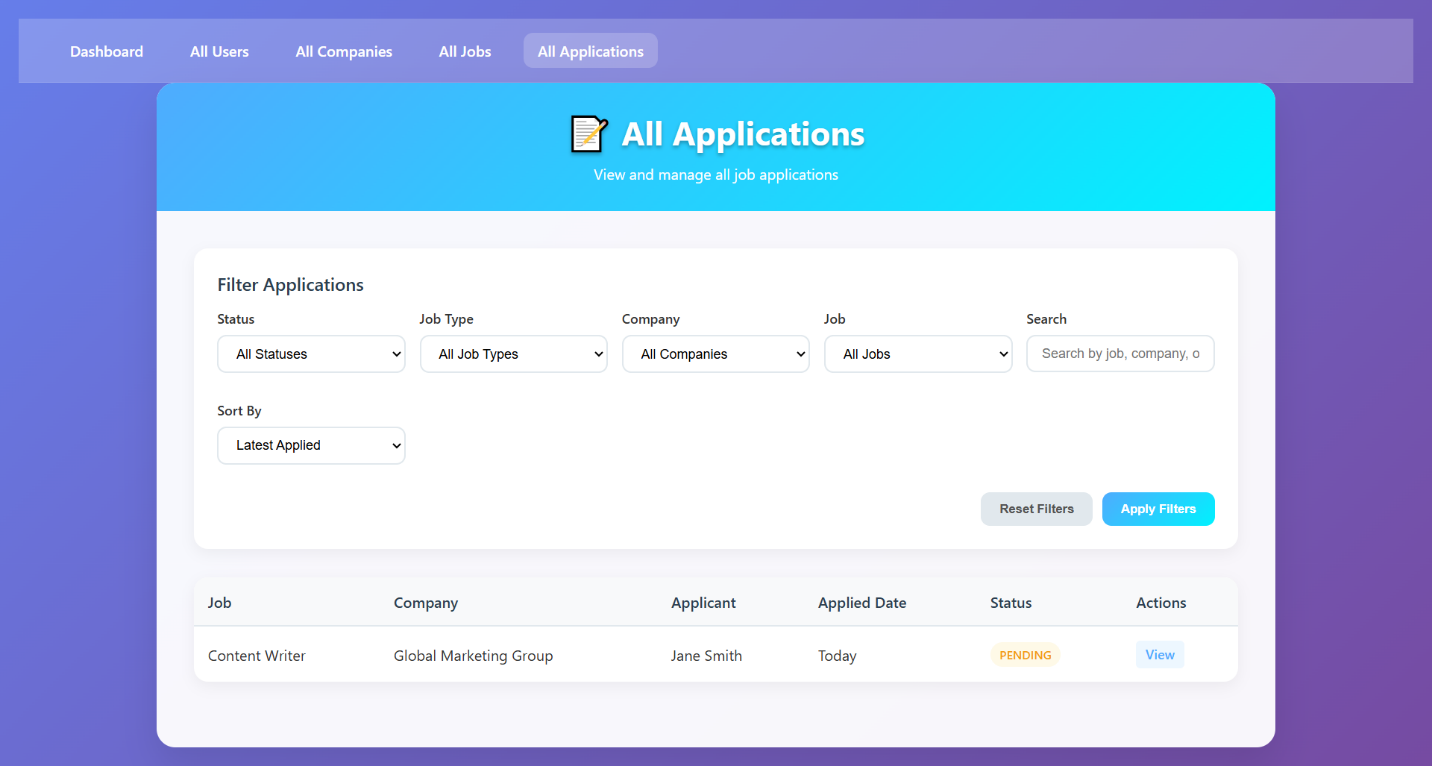
**Job Details:**

****

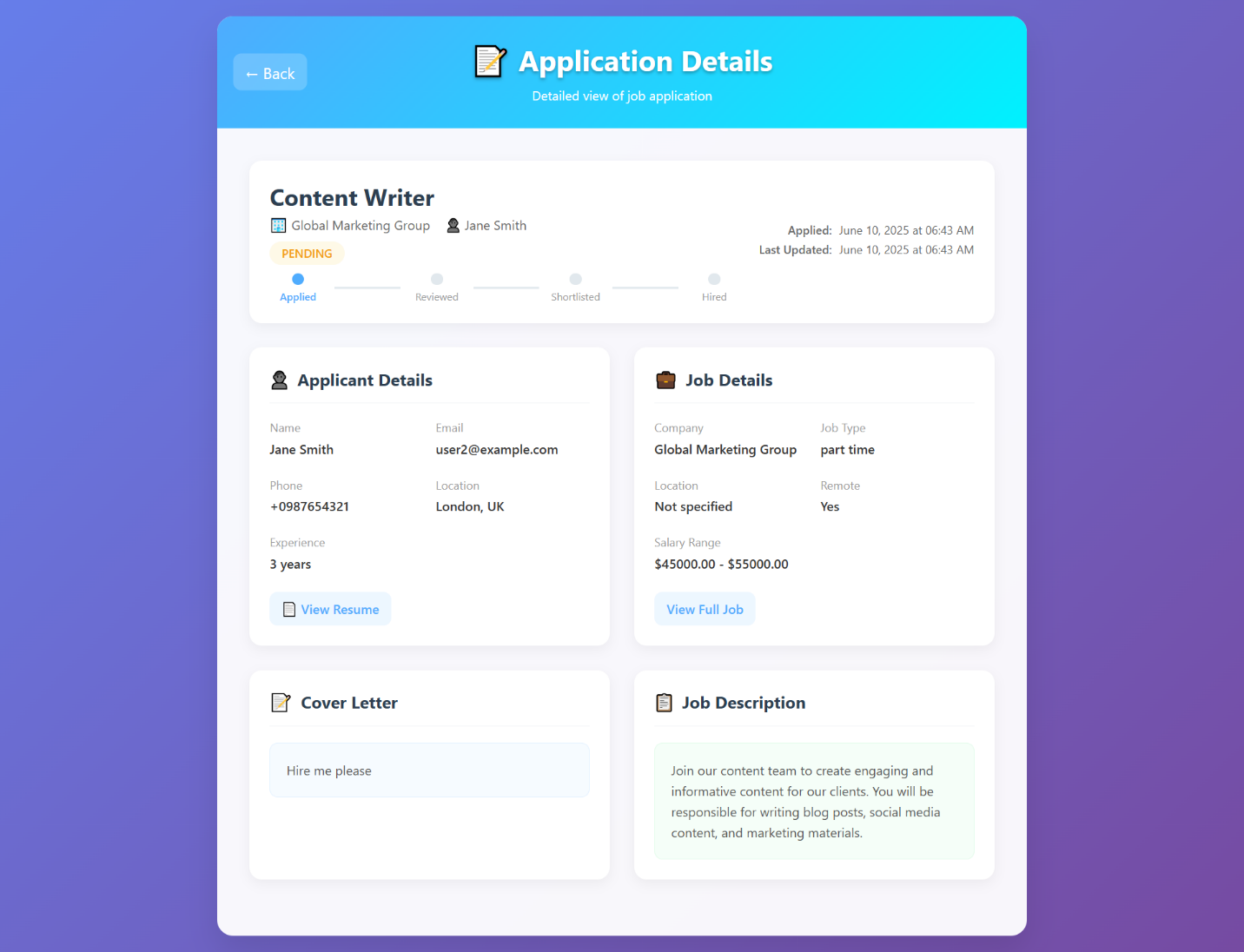
**Apply For Job:**

****

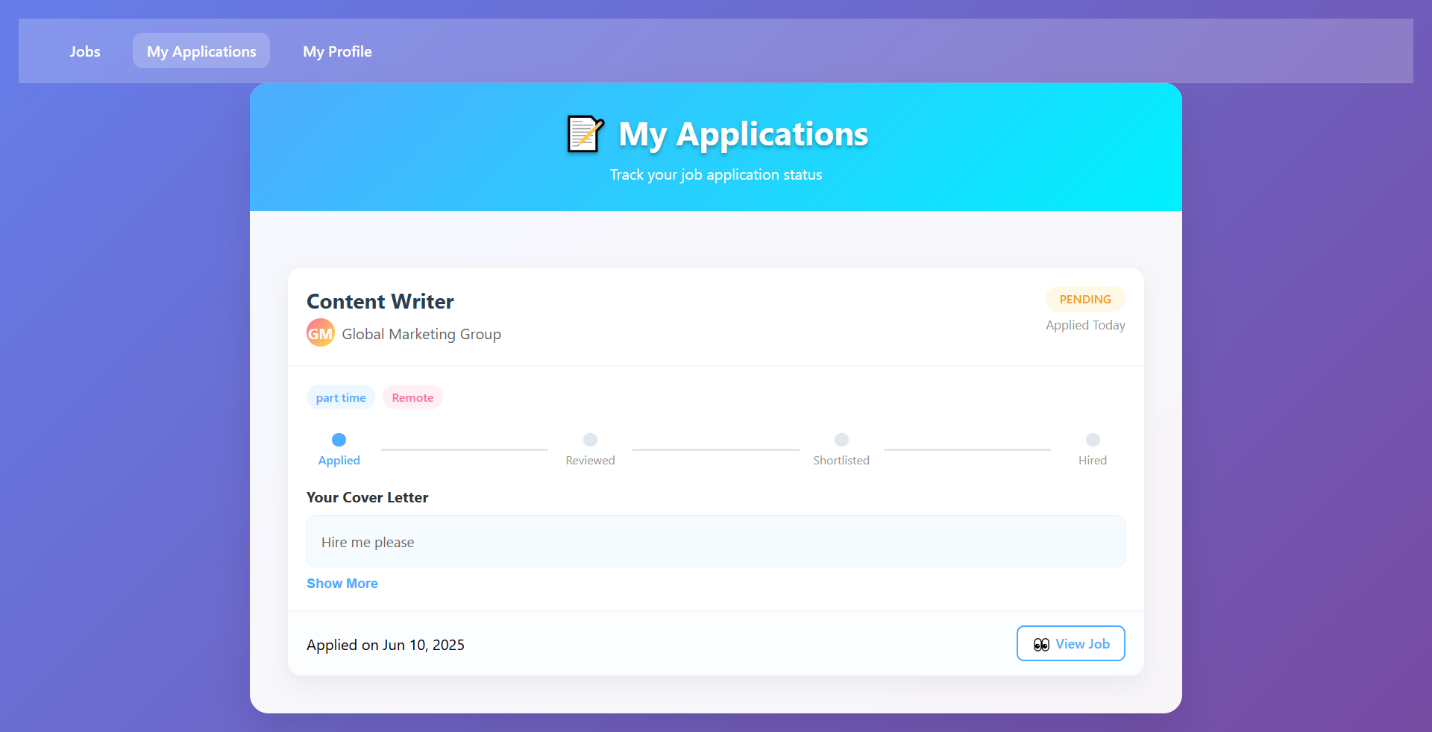
**All Applications:**

****

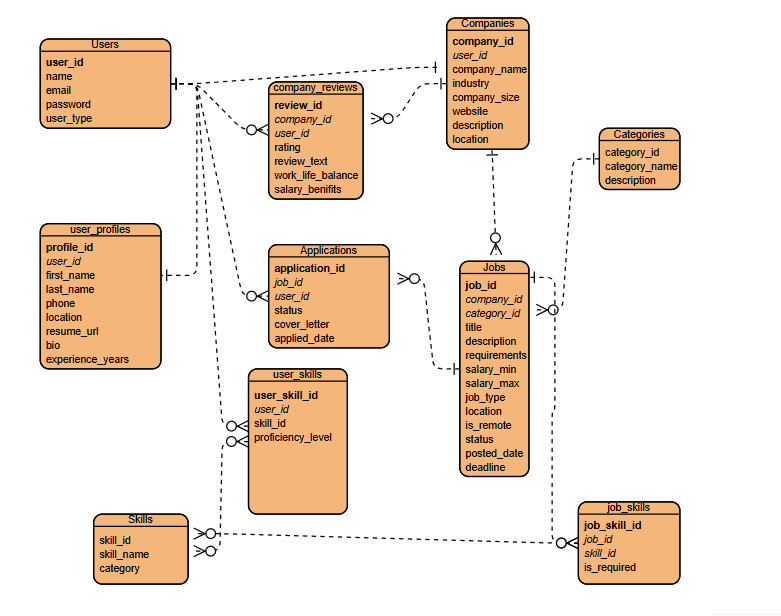
**Single Application:**

****

**My Application:**

****

***ERD***



**Thank You**