

# Database-Driven Talent Matching Platform

## Project Proposal Report & ER Diagram

### a. Description:

This project focuses on developing a database-centered web application that connects vetted software engineers with companies seeking skilled developers through structured data management and intelligent query-based matching. Operating in the Recruitment domain, the system emphasizes relational database design to manage complex user interactions, such as engineer vetting, job postings, and match generation. With efficiency and data integrity. The platform's MVP will include three main roles: Engineers, Clients, and Administrators. Engineers can create profiles and apply for vetting, clients can post job requirements, and the system retrieves best-fit candidates using optimized database queries. While the project incorporates a basic automated matching algorithm, the primary focus is to demonstrate how a well-designed relational database schema can power an entire multi-role marketplace application with real-world functionality.

### b. Problem Statement Motivation

The technology industry continues to face persistent challenges in efficiently connecting qualified developers with companies that require their skills. Hiring processes are often slow, disorganized, and heavily dependent on manual screening, while existing online recruitment platforms struggle with unstructured data management and limited transparency in how candidates are matched to roles.

To address these issues, we plan to develop a centralized, database-driven recruitment platform that organizes and links all relevant hiring information, including engineer profiles, skills, job postings, and vetting records, within a well-structured relational database. By using optimized SQL queries, relational joins, and integrity constraints, the system will generate efficient and accurate candidate–job matches directly from the database. This approach will streamline data retrieval, reduce redundancy, and ensure that every hiring action is traceable and consistent. Through this structured data model, our platform will make the overall recruitment process faster, more transparent, and more reliable for both companies and developers.

### c. Objective and Major System Features

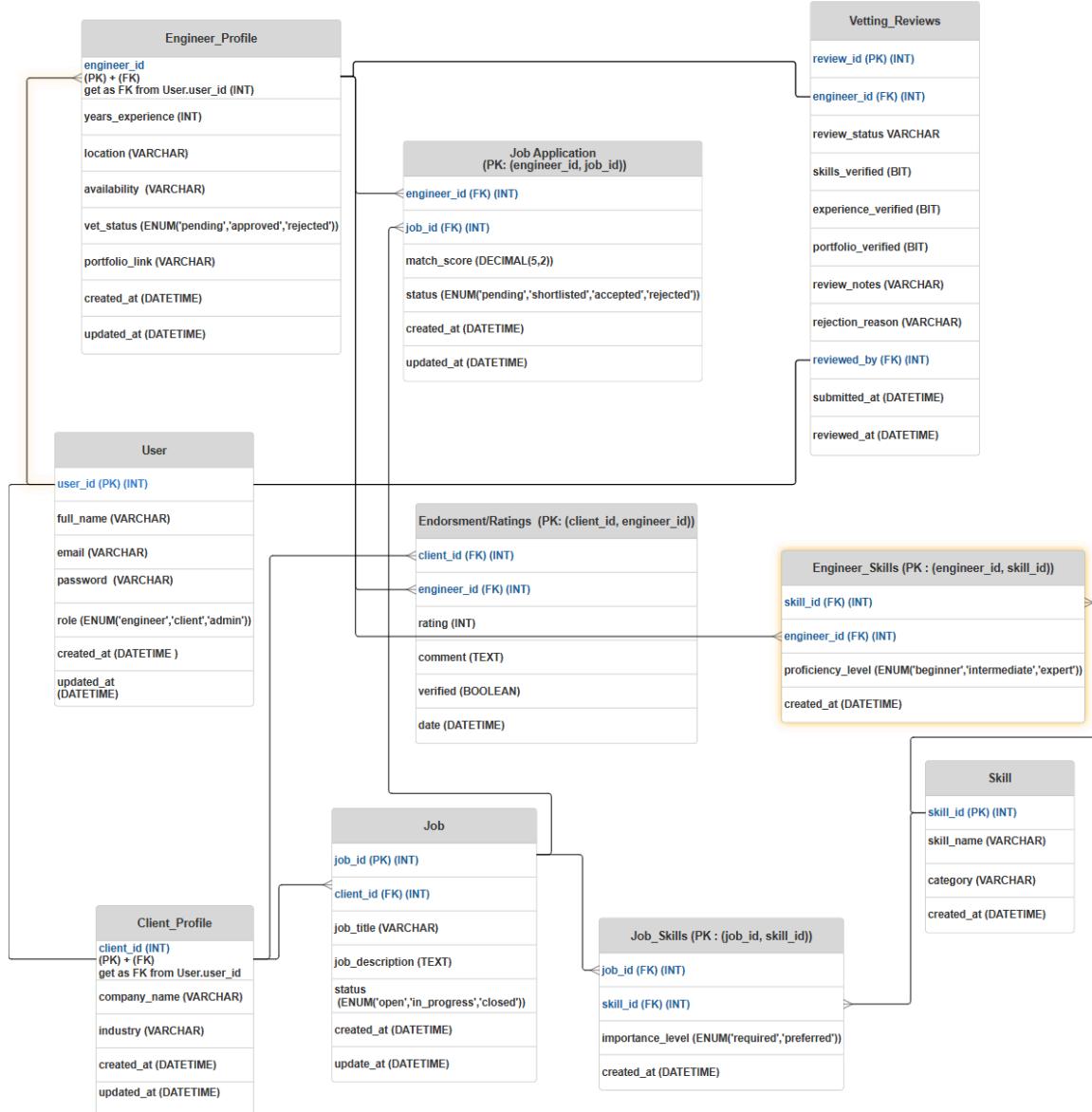
#### Objectives

1. To design and implement a normalized relational database that effectively represents real-world recruitment data (users, skills, jobs, and vetting statuses). We'll achieve this through entity-relationship modeling and 3-NF normalization
2. To enable efficient data retrieval and candidate job-matching using advanced SQL queries and indexing strategies to optimize performance and make the project scalable.
3. To ensure database integrity and consistency through enforcement of primary/foreign key constraints, validation rules and role-based access control
4. Integrate the database with a .NET frontend using Entity Framework, allowing the system to switch between LINQ queries and stored procedures at runtime with the help of the Factory Pattern.
5. To simulate a real-world hiring process where SQL Server queries and relational joins drive all key operations (vetting, job posting, and candidate matching), making sure that application logic is rooted in fundamentally well-designed database interactions
6. To highlight how structured data design can replace unorganized, ad hoc methods of talent sourcing.

## Major System Features

- **Engineer Module:**  
Engineers will be represented through dummy data in the current phase to simulate real user profiles. Each engineer record will include key details such as skills, experience proficiency level, years of experience, location, and availability. This data will be stored across normalized tables like EngineerProfile and EngineerSkill, allowing the team to test relational queries and matching logic before integrating actual user registration functionality.
- **Client Module:**  
Clients will also be represented through dummy data to simulate job postings. Each job record will include details such as required skills, experience level, and preferred time zone. Job information will be stored in tables such as Job and JobSkill, allowing efficient filtering and candidate matching.
- **Database-Based Matching Engine:**  
Our system will rely on SQL-based logic to identify suitable engineers. Through relational joins and match scoring (based on skills, experience, and availability), the database will produce ranked candidate results for each job posting.
- **Admin Dashboard:**  
Administrators will manage user roles, review and approve engineer applications, and oversee the database for data accuracy and system integrity. Access will be controlled through role-based permissions implemented at the database level.
- **Database Schema & Integrity Controls:**  
The database will be the project's core component, featuring entity-relationship modeling, normalization (up to 3NF), foreign key enforcement, and indexing for optimized queries. These design choices will ensure scalability, efficiency, and data consistency across all user operations.

# Entity-Relationship Diagram



Names	Roll numbers
Muhammad Shazil Nadeem	27100183
Taimur Jahanzeb	27100028
Haider Wajahat	27100252
Syed Zaid Ahmed	27100192
Areeba Naveed	27100239