**Backend Developer Assignment: Company ERP User Management System**

**Duration**: 4 Days  
**Tech Stack**: Express.js or NestJS (Your Choice) + MySQL

### 🎯 Objective

Design and build a backend module for an ERP platform to manage users within companies. Your task is to create a secure, multi-tenant system where each company can manage its internal users with different roles.

### 🧩 Background

* Companies and their respective roles are created manually by the backend developer in the database.
* Each company will have an initial user (with the CA role) who will be created manually in the database.
* This CA (Company Admin) will log in via the ERP and can create other users belonging to their company (e.g., Store Managers, Finance Managers).
* All users belong to only one company.
* Role-based access should be implemented.

### ✅ What You Need to Implement

1. **Login API (POST /auth/login)**
   * Accepts email and password.
   * Authenticates user and returns JWT containing user ID, company ID, and role ID.
   * Passwords must be hashed using bcrypt.
2. **Create User API (POST /users)**
   * Only accessible by logged-in users with the CA role.
   * Allows the CA to create other users within their own company.
   * The created user should inherit the company ID of the logged-in CA.
   * created\_by should store the ID of the CA creating the user.
3. **Get All Users (GET /users)**
   * Only accessible to logged-in users.
   * Returns a list of users belonging to the same company.
   * Support pagination.
4. **Get All Roles (GET /roles)**
   * Returns the list of available roles.
   * These roles are already defined in the system by the backend team.
5. **GET /me**
   * Return current user’s profile based on the JWT token.

### 🔐 Auth & Security

* Use JWT for authentication.
* All protected routes must decode JWT and ensure data access is restricted to the user’s company.
* Passwords must be securely hashed.

### 🔥 Bonus (Optional)

* Add soft-delete support (e.g., is\_deleted flag on users).
* Write basic test cases (Jest/Mocha).

### 📝 Submission Guidelines

* Push the full project to a **public GitHub repository**.
* Include a detailed **README.md** with:
  + Setup instructions
  + How to run and test the project
  + Database connection setup
  + Sample .env file (include in repo)
* Ensure we can **clone and run the project locally** without issues.

### 🔧 Hints

* Think about what kind of structure the company, role, and user data should follow.
* Do not hardcode role or company values — assume they already exist.
* Follow best practices in organizing your backend code.

Good luck! We’re looking forward to your submission.