💼 Backend Developer Job Task

—--***----------—---------------------------------

Submit to: fsdteam.saa@gmail.com

—--***----------—---------------------------------

"HireMe – Job Posting Platform"

📘 Scenario:

You are tasked with building the backend system for **HireMe**, a job platform where companies can post jobs and job seekers can apply. Applying for a job requires a **payment of 100 Taka(Optional)**. The system must support **role-based access control** to manage permissions for Admins, Employees (recruiters), and Job Seekers.

🧩 Roles & Permissions

Role	Permissions
Admin	Manage all users (create/update/delete), manage jobs, see company analytics
Employee(recruiters)	Post/edit/delete jobs for their company, view applicants, accept/reject them
Job Seeker	View jobs, apply by uploading CV + paying 100 Taka

---

✅ Functional Requirements

🔐 Role-Based Authentication

- Use **JWT-based authentication**.
- Must implement **role checks** for protected routes.

- Only Admins can manage users.
- Employees can only manage their own jobs and applications.
- Job Seekers can apply only **after payment**.

📁 File Upload

- Job Seekers must upload their **CV/resume** (PDF, DOCX).
- Use **Multer** (either Disk or Memory storage).
- Validate file type and size (max 5MB).

💳 Payment System

- Integrate a **fake/mock Stripe/SSLcommerz** (or optional real Stripe if you want to go further).
- Job Seekers must **pay 100 Taka** to apply for a job.
- After successful payment:
    - Save application with payment status.
    - Store a basic invoice object (id, user, amount, time).

🧠 Admin Panel (Backend-Only)

- Admins can:
    - View all users, jobs, applications.
    - Filter by company or status.

- Employees(recruiters) can:
    - View applications to their jobs.
    - Accept or reject applications.

- Job Seekers can:
    - View job listings.
    - Apply for jobs they haven't already applied to.
    - View their application history.

🛠️ Tech Requirements

- **Language**: JavaScript or **TypeScript (optional)**
- **Framework**: Express.js
- **Database**: MongoDB or PostgreSQL
- **Authentication**: JWT
- **File Upload**: Multer
- **Payment**: Stripe or mock service or others
- **Validation**: Zod or Joi (optional but recommended)
- Environment configs via .env

💡 Bonus Points

- Proper folder structure (routes, controllers, middlewares, services, etc).
- Use TypeScript with interfaces/types.
- Input validation using Zod.
- Use of enums/constants for roles and statuses.

📦 Deliverables

- **GitHub repo** (public) with clean code and README.
- **README** must include setup, API endpoints, roles, and payment flow.
- **ERD** (PDF or image format in the repo).
- **Hosted API (optional)** using Render, Railway, etc.
- **Postman** documentation(Must)

Best of luck!

NOTE:

—----------***----------—----------

Submit to: fsdteam.saa@gmail.com

—----------***----------—----------