Final HRMS Backend Documentation (Roman Urdu) - Bilal Raza

1. Packages aur Unka Kaam

express - Node.js backend framework (routes banane ke liye)

mongoose - MongoDB se connect hone ke liye aur schema banane ke liye

dotenv - .env file se secrets access karne ke liye

bcryptjs - Password ko hash karne ke liye

jsonwebtoken - JWT token banane aur verify karne ke liye (auth system)

nodemailer - Email bhejne ke liye (invite system)

multer - File upload (image/resume) ke liye

cloudinary - Uploaded file ko online Cloud me store karne ke liye

cors - Frontend aur backend ke beech me link banane ke liye

express-async-handler - Async errors handle karne ke liye

cookie-parser - Cookies (JWT) ko read karne ke liye

2. Folder Structure (Aur Har Folder Ka Kaam)

config/ => db.js, cloudinary.js (Database & cloud setup)

controllers/ => Har route ka kaam yahaan likha jata hai (logic)

middleware/ => JWT check, role check, error handler

models/ => Saare schema yahan honge (Company, User, Request, Owner)

routes/ => API routes define hote hain (companyRoutes, authRoutes etc.)

utils/ => Reusable functions (e.g. sendEmail)

uploads/ => Temporary local file store (optional)

.env => Secrets jaise DB_URI, JWT_SECRET

server.js => Entry point (app start hota hai yahaan se)

3. Saare Schemas (Alag Alag)

Company.js:

- name, email, slug, logoUrl, etc.
- users: [embedded users]
- applications: [public job applicants]
- createdAt, updatedAt

User (embedded in company):

- name, email, password, role (admin/hr/employee)
- isActive, reg_code, departmentId (optional)
- profilePic, salary, gender, dob, joiningDate
- documents[], attendance[], leaves[], salaryHistory[]
- interviewStatus

Request.js:

- companyName, email, phone, message, status, createdAt

Owner.js:

- name, email, password (admin of whole product)
- role: 'owner'
- Isko manually DB me insert karna hai, koi route nahi hoga

4. Saare Routes Aur Kaam (with Middleware & Controller)

POST /api/requestService

- Public form
- No middleware
- Controller: requestController.createRequest()

POST /api/inviteCompany

- Owner only
- Middleware: protect, isOwner
- Controller: companyController.inviteCompany()

POST /api/registerCompany

- Public invite link (slug ya token se)
- No middleware
- Controller: companyController.registerCompany()

POST /api/registerAdmin

- Public form after company register

- Controller: companyController.registerAdmin()

POST /api/:company/login

- Public

- Controller: authController.loginUser()

GET /api/:company/dashboard

- Middleware: protect, role check

- Controller: companyController.getDashboard()

POST /api/:company/attendance

- Middleware: protect, isHRorAdmin

- Controller: userController.markAttendance()

POST /api/:company/leaveRequest

- Middleware: protect

- Controller: userController.requestLeave()

POST /api/:company/inviteUser

- Middleware: protect, isAdmin

- Controller: companyController.inviteUser()

DELETE /api/:company/user/:email

- Middleware: protect, isAdmin

- Controller: companyController.fireUser()

5. Step-by-Step Backend Roadmap (Beginner Friendly)

STEP 1: npm init + saare packages install karo (section 1 ke mutabiq)

STEP 2: server.js banao + express, middleware, DB connect setup karo

Final HRMS Backend Documentation (Roman Urdu) - Bilal Raza

STEP 3: config/db.js aur .env file ready karo

STEP 4: models/ folder me saare schema create karo (Company.js, Request.js, Owner.js)

STEP 5: middleware/ folder me protect.js, roleCheck.js, error.js banao

STEP 6: routes/ aur controllers/ folder me basic routes likhna start karo (request, registerCompany, login)

STEP 7: JWT logic banao (login & protect)

STEP 8: Multer + Cloudinary setup karo (logo, profile pic, resume upload ke liye)

STEP 9: Email system setup karo (utils/sendEmail.js)

STEP 10: Invite links, admin register, attendance, leave routes banao

STEP 11: POSTMAN se saari APIs test karo step by step

STEP 12: Owner ka account manually DB me daal do (role: 'owner')

6. Extra Secure Points (Important!)

- Kisi bhi user ka password kabhi frontend pe mat bhejna
- company.users[] ko bhejne se pehle sensitive fields hata do (password, salary)
- Auto login after register: JWT set karo aur user ko dashboard redirect karo
- Slug use karo unique URL ke liye (e.g. the-big-company-xyz123)
- Company name show karte waqt original name use karo, slug sirf URL ke liye