📘 HR Management System – Full Project Scope (Advanced Version)

🧠 Objective:

Design and develop a **complete HRMS** using **MERN Stack** (MongoDB, Express.js, React.js, Node.js) to manage **multiple companies**, **employee records**, **salaries**, **attendance**, and **HR operations**.

👤 User Roles:

Role	Description
Owner	Creates and manages companies, invites company admins
Company Admin	Manages company settings, HR users, and employee data
HR	Manages employees, attendance, salary, and leave records
Employee	Can view their profile, attendance, salary sheet, apply for leave, etc.

🖥️ Frontend Modules (React.js)

🔐 Authentication:

- Role-based login
- Email verification / reset password
- Invite-based employee onboarding with reg_code

🧾 Employee Application & Profile:

- Complete application form (with CV upload)
- Auto-populated fields for invited users
- Editable user profile for employees

📊 Admin/HR Dashboard:

- Overview (employees count, attendance today, pending approvals)
- Tab-wise access: `Employees`, `Attendance`, `Salary`, `Leave Requests`, `Interview Schedules`

📁 Modules:

1. 👨‍💼 Employee Management

- Add/Edit/Delete employee
- Assign departments and roles
- View employee profile (contact, DOB, address, CV, etc.)

2. ⏰ Attendance Management

- Mark daily attendance (Manual / Auto)
- Show present/absent summary per employee
- Calendar view
- Filter by month, department, employee

3. 💵 Salary Management

- Add base salary

- Add allowances, bonuses, and deductions
- Auto-generate monthly salary sheet
- Download/export salary reports (PDF/CSV)
- Maintain salary history
- Approval system for final payment

4. 🏝️ Leave Management

- Employee can apply for leave
- HR/Admin approves/rejects leave
- Track leave days per employee
- Leave types: Casual, Sick, Annual, etc.

5. 🗓️ Interview Scheduling

- Company admin can schedule interview (date, time, location)
- Track candidate status: Applied → Scheduled → Interviewed → Hired/Rejected

---

🔧 Backend (Express.js + MongoDB)

📌 Schemas Design

`User`:

```
{
  name,
  email,
  phone,
  gender,
  dob,
  address,
  role, // owner, admin, hr, employee
  companyId,
  departmentId,
  reg_code,
  isActive,
  createdAt
}
```

`Company`:

```
{
  name,
  createdBy, // Owner
  admins: [userIds],
  hrUsers: [userIds]
}
```

`Attendance`:

```
{
  employeeId,
  companyId,
  date,
  status: ['present', 'absent', 'leave', 'holiday'],
  checkInTime,
  checkOutTime
}
```

`Salary`:

```
{
  employeeId,
  companyId,
  month,
  year,
  baseSalary,
  bonuses,
  deductions,
  finalSalary,
  status: ['pending', 'approved', 'paid'],
  generatedAt
}
```

`LeaveRequest`:

```
{
  employeeId,
  fromDate,
  toDate,
  reason,
  type: ['casual', 'sick', 'annual'],
  status: ['pending', 'approved', 'rejected'],
  reviewedBy
}
```

`InterviewSchedule`:

```
{
  candidateId,
  companyId,
  interviewerId,
  date,
  time,
  location,
  status: ['scheduled', 'done', 'rejected', 'hired'],
  notes
}
```

🧠 Business Logic Highlights

🗓️ Attendance Auto-Population:

- Automatically mark *"Absent"* for employees who haven't checked in by a cut-off time.
- Allow manual override for correction by HR/Admin.

📉 Salary Sheet Generation:

- Every month, calculate:
  - Base salary
  - Bonus (if any)

- Deductions (leave without pay, penalties)
- Final amount

- Allow HR/Admin to approve and mark as *Paid*
- Generate downloadable reports per employee/company

📑 Leave Deductions:

- If the leave quota is exceeded, auto-deduct from salary

🛡️ Security & Access Control

Feature	Owner	Admin	HR	Employee
View Companies	✅	❌	❌	❌
Manage Employees	❌	✅	✅	❌
View Own Profile	❌	❌	❌	✅
View/Generate Salary	❌	✅	✅	✅ (own only)
Attendance Control	❌	✅	✅	❌
Apply Leave	❌	❌	❌	✅
Approve Leave	❌	✅	✅	❌

✅ Features for Completion

MVP (Minimum):

- Company Creation (Owner)
- Admin Invitation

- Employee Application Form
- Role-based Login
- Attendance Sheet
- Basic Salary Sheet (Manual entry)
- CV Upload

Bonus:

- Interview Scheduling
- Leave System
- Salary Auto Calculation
- CSV/PDF Export
- Email Notification

🎓 Student Tasks Breakdown

Module	Responsibility
Auth System	All students
Company/Admin Setup	1-2 students
Employee Form & CV Upload	2 students
Attendance System	2 students
Salary Module	2-3 students
Leave & Interview System	Optional/Bonus
Role-based Access & Layouts	All

🔚 Final Deliverables

1. MERN App with proper folder structure
2. Seed user: Owner
3. At least 2 companies with users
4. Realistic employee data
5. Working salary sheet
6. Attendance for 1 month
7. Salary calculation logic
8. Leave requests + basic HR approval
9. Neat UI (preferably with Material UI or Tailwind)
10. Code should be modular, clean and well-commented

Here Some Technical Guidance

1. ✅ Required **API endpoints** (Backend)
2. ✅ Recommended **MongoDB collections (tables)**
3. ✅ Instructions for **displaying user data**
4. ✅ Guidelines for **editing user info via modals**

🔗 API Endpoints (Express.js – RESTful Design)

All endpoints are protected via authentication and role-based middleware unless marked as public.

🔐 Auth Endpoints

Method	Endpoint	Description
`POST`	`/api/auth/register`	Register a new user (employee via invite)
`POST`	`/api/auth/login`	Login with email & password
`POST`	`/api/auth/invite`	Invite a user by generating a `reg_code`
`POST`	`/api/auth/reset-password`	Initiate/reset password flow

🏢 Company & User Management

Method	Endpoint	Description
`POST`	`/api/companies`	Create a new company (Owner only)
`GET`	`/api/companies`	List all companies (Owner only)
`POST`	`/api/users`	Create employee/admin/HR (Admin only)
`GET`	`/api/users`	List all users in a company
`GET`	`/api/users/:id`	Get specific user details
`PUT`	`/api/users/:id`	Update user details
`DELETE`	`/api/users/:id`	Deactivate or delete user

📆 Attendance Endpoints

Method	Endpoint	Description
`POST`	`/api/attendance/mark`	Mark today's attendance (HR/Admin)
`GET`	`/api/attendance/user/:id`	Get attendance for a user
`GET`	`/api/attendance/company`	Get attendance report for company

💰 Salary Sheet Endpoints

Method	Endpoint	Description
`POST`	`/api/salary/generate`	Generate salary for an employee
`GET`	`/api/salary/user/:id`	Get salary history of employee
`GET`	`/api/salary/company`	Get salary sheets per department or month
`PUT`	`/api/salary/:id/status`	Mark salary as paid/approved

🏝️ Leave Endpoints

Method	Endpoint	Description
`POST`	`/api/leave/request`	Employee applies for leave
`GET`	`/api/leave/company`	Get all leave requests (Admin/HR)
`PUT`	`/api/leave/:id/status`	Approve/Reject leave

🧾 Interview Scheduling

Method	Endpoint	Description
`POST`	`/api/interview/schedule`	Schedule interview
`GET`	`/api/interview/company`	Get all scheduled interviews

🧩 MongoDB Tables (Collections)

Collection	Purpose
`users`	Stores all users (owner, admin, HR, employee)
`companies`	Stores company info and their owner/admins
`attendance`	Stores daily attendance of employees
`salaries`	Stores salary sheets and payment status
`leaveRequests`	Tracks leave applications
`interviews`	Stores interview schedules

📋 Displaying User Data

On Admin Dashboard → "Employee Management" Table

Each row includes:

Name	Email	Role	Department	Status	Actions
John Doe	john@email.com	HR	Sales	Active	🖊️ Edit

Use MUI, AntD, or Bootstrap Table with pagination and search filters.

✏️ Edit Modal for User Info

When the **Edit button** is clicked:

- Open a modal (popup)
- Load current user data into a form
- Allow admin to update editable fields

Example Fields in Modal:

```
<Form>
  <Input label="First Name" defaultValue={user.firstName} />
  <Input label="Last Name" defaultValue={user.lastName} />
  <Input label="Phone" defaultValue={user.phone} />
  <Select label="Role" options={['HR', 'Employee']} />
  <Select label="Department" />
  <Button onClick={handleSubmit}>Save</Button>
</Form>
```

On submit:

- Validate data
- Make `PUT /api/users/:id` request
- Show success toast and refresh table

🧠 Bonus Suggestions

- Use **React Query** or **Redux Toolkit Query** for fetching/updating data
- Add inline **toast messages** for better UX
- Keep modals modular & reusable for different types (user, salary, leave)