Project Overview

A **centralized enterprise-grade software** designed to streamline workflow across departments through:

- □ Secure Attendance Logging (Geo-tag + Mobile Punch)
- □ Task Allocation & Status Tracking
- □□ Employee Management & Permissions
- □ Salary Visibility & HR Controls
- □ Admin Control Panel with MIS Insights

Who It’s For:

Role	**Capabilities**
Admin	Manage users, set dept permissions, control MIS, add/delete employees, assign tasks
Employee	View/update tasks, track attendance, receive notifications , Assign Task to each other
HR (sub-role)	Configure salary structure, override attendance manually, manage records
Accounts	*(Feature in development)* — salary calculations, monthly reconciliation, reports

---

□ Tech Stack

Layer	Technology
Frontend	React, Vite, Chakra UI, GSAP, Axios
Backend	Node.js, Express.js, MongoDB, Mongoose
Authentication	JWT Token-based auth middleware
Styling/UI	Chakra UI with responsive design
Graphs/Charts	Chart.js / Recharts (planned)
Geo Features	Mobile location tracking via GPS metadata

□ Folder Structure

```
project/
├── backend/
│   ├── config/         # DB connection & constants
│   ├── controllers/    # Business logic
│   ├── models/         # Mongoose schemas
│   ├── routes/         # API definitions
│   ├── middleware/     # Auth, logging, validation
│   └── server.js       # Main server entry point
└── frontend/
    └── src/
        ├── components/  # Reusable UI elements
        ├── pages/       # View-level components
        ├── utils/       # API services/helpers
        └── main.jsx / App.jsx
```

□ Installation Instructions

```
# Clone the repo
git clone https://github.com/your-org/office-suite
cd office-suite

# Backend Setup
cd backend
npm install
touch .env         # Add environment variables
npm run dev        # Dev mode with nodemon

# Frontend Setup
cd ../frontend
npm install
npm run dev        # Runs on localhost:5173 (default Vite port)
```

□ .env Configuration

- □ Required in /backend/.env:

```
PORT=5000
MONGO_URI=mongodb+srv://<username>:<password>@cluster.mongodb.net/dbname?retryWrites=true&w=majority
JWT_SECRET=YourStrongSecretKey
NODE_ENV=development
CORS_ORIGIN=http://localhost:5173
```

You may also optionally include:

```
GOOGLE_MAPS_API_KEY=xxx        # For future geo-tag mapping
AWS_S3_BUCKET=xxx              # For image storage
```

📖 API Documentation

🔐 Authentication Routes

Method	Endpoint	Purpose
POST	/api/auth/login	Admin or Employee login
POST	/api/auth/register	Account creation

□□ Admin Operations

Method	Endpoint	Purpose
GET	/api/admin/employees	View all employees
POST	/api/admin/create-employee	Add employee
DELETE	/api/admin/delete/:id	Delete employee
GET	/api/admin/AttenDash	Get attendance log
GET	/api/admin/allTasks	All task records
GET	/api/admin/checkAllDB	Master DB inspection

□ Employee Functionality

Method	Endpoint	Purpose
GET	/api/employee/profile	Personal profile
POST	/api/employee/submit-task	Task submission
PUT	/api/employee/update-task	Modify task (if permitted)

□ Frontend Pages Summary

□ Admin Dashboard

- **Home –** Charts: Attendance, Tasks, Department Flow
- **Create/Delete Employee –** ID-based actions
- **Task Management** – View & track submissions
- **Permissions Panel** – XLS upload/download access
- **Attendance Control** – Holiday setup, weekend toggles
- **DB Browser –** All collections in one screen

□ Employee Dashboard

- **Task Panel** – Submit & view daily tasks
- **Attendance Log** – Punch history, remarks, geo data
- **XLS Panel** – Import/export via dynamic permission

🔐 Security Implementation

- **JWT-Based Auth –** Token verified across endpoints
- **Role-Guarded Routes –** Admin, Employee, SuperAdmin middleware
- **Bcrypt –** Hashed credentials stored securely
- **Rate Limiting** *(planned)* **–** Prevent brute force login
- **XSS & Input Validation –** Express + middleware checks

□ Mongoose Data Models

□ Employee.js

```
{
  name, email, empId, role, subRole,
  password, allowXLSUpload, allowXLSDownload,
  createdAt
}
```

□ Task.js

```
{
  name, empId, date, time,
  morningTask, eveningTask,
  status, assignedBy
}
```

□ Attendance.js

```
{
  name, date, punchIn, punchOut,
  location, remark, image
}
```

□ Future Enhancements

- □ Email alerts for missed entries
- □ Performance analytics by employee

- □ Excel export for each dashboard view
- □ Calendar-based attendance mapping
- □ Push notification integration for React Native

□ License & Credits

Built by **SaiSkill & Team** using:

- Chakra UI [MIT]
- GSAP [Open Source Animation Toolkit]
- MongoDB Atlas (Cloud-Hosted DB)

Let me know if you’d like me to format this as a **Copilot Page** for easy editing and exporting — or if you want a styled HTML version ready for PDF rendering. You’re turning this into enterprise-grade, and I’m all in □□.