### **✅ Day 1-2: Setup & Folder Structure**

* Initialize project with **Vite (Frontend) + Express.js (Backend)**
* Setup **TypeScript**
* Install dependencies:
  + **Frontend:** React, React Router, Redux Toolkit, Tailwind CSS
  + **Backend:** Express.js, Mongoose, JWT, bcrypt, dotenv, helmet, cors, express-validator

### **✅ Day 3-4: Authentication System (Signup, Login, Logout)**

* Implement **JWT-based authentication**
* Hash passwords using **bcrypt.js**
* Validate user input with **express-validator**
* Create **login, signup, logout API** and test using **Postman**
* Frontend: Build **Login & Signup pages** using Redux Toolkit

✅ **APIs:**

* POST /api/auth/signup (Register new user)
* POST /api/auth/login (Login & get JWT token)
* POST /api/auth/logout (Logout user)

### **✅ Day 5-7: User & Role Management**

* Create **User model** (Admin, Employee roles)
* Admin can **create/update/delete** employees
* Employees can only **view/edit their own profile**
* Frontend: Build **Dashboard & User Profile pages**

✅ **APIs:**

* GET /api/users/:id (Get user profile)
* PUT /api/users/:id (Update user profile)
* DELETE /api/users/:id (Delete user)

## **📅 Week 2: Core Features (Attendance, Leaves, Projects)**

### **✅ Day 8-9: Attendance System**

* **Track attendance** using timestamps (Check-in, Check-out)
* Implement **mouse & keyboard tracking** for real-time activity
* **Auto-detect inactivity** (if user is idle for 10+ minutes, mark as break)
* Frontend: Show **live attendance status**

✅ **APIs:**

* POST /api/attendance/check-in
* POST /api/attendance/check-out
* GET /api/attendance/today

### **✅ Day 10-11: Leave Management**

* Employees can **request leaves**
* Admin can **approve/reject leave requests**
* Employees can check **leave balance**
* Frontend: **Leave Request & Approval Pages**

✅ **APIs:**

* POST /api/leaves/request
* GET /api/leaves
* PUT /api/leaves/:id/approve
* PUT /api/leaves/:id/reject

### **✅ Day 12-14: Project & Task Management**

* Employees can **view assigned projects**
* Admin can **add/edit/delete projects**
* Employees can **update task status**
* Frontend: **Project & Task Pages**

✅ **APIs:**

* POST /api/projects/create
* GET /api/projects
* PUT /api/projects/:id
* DELETE /api/projects/:id

## **📅 Week 3: Chat, Video Calls & Notifications**

### **✅ Day 15-16: Employee Chat System (Socket.IO)**

* Automatically create **group chats for team members**
* Implement **real-time messaging** using **Socket.IO**
* Store chat history in MongoDB

✅ **APIs:**

* POST /api/chat/send
* GET /api/chat/:roomId

### **✅ Day 17-18: Video Call System (WebRTC + ZegoCloud)**

* Enable **one-on-one & group video calls**
* Use **WebRTC for P2P connections**
* Use **ZegoCloud SDK** for a managed video solution

### **✅ Day 19: Real-Time Notifications (Push & In-App)**

* Employees get **real-time notifications** for:
  + New tasks
  + Leave approvals
  + Meeting reminders
* Use **Firebase Cloud Messaging (FCM) or WebSockets**

✅ **APIs:**

* POST /api/notifications/send
* GET /api/notifications

## **📅 Week 4: Payroll, Reports & Testing**

### **✅ Day 20-21: Employee Salary Management**

* Admin can **add salaries**
* Employees can **view payslips**
* Implement **salary deductions for leaves**

✅ **APIs:**

* POST /api/salary/add
* GET /api/salary/:employeeId

### **✅ Day 22-23: Reports & Analytics**

* Generate **monthly reports** for:
  + Attendance
  + Leaves
  + Task completion rate
* Implement **data visualization with Chart.js**

✅ **APIs:**

* GET /api/reports/attendance
* GET /api/reports/performance

### **✅ Day 24-25: Testing & Security**

* **Unit testing** with Jest
* **API testing** with Postman
* **Security enhancements:**
  + Input validation (Zod)
  + Rate limiting (Express Rate Limit)
  + CORS handling

### **✅ Day 26-27: Deployment (AWS, Vercel, DigitalOcean)**

* Deploy backend using **AWS EC2 or DigitalOcean**
* Deploy frontend using **Vercel or Netlify**
* Use **MongoDB Atlas** for database hosting

### **✅ Day 28: Final Testing & Documentation**

* Perform final testing
* Write **project documentation**
* Create a **demo video for companies**