

Employee Attendance System – Project Explanation

1. Overview

This is a full-stack web application that manages employee attendance with two roles:

- Employee – marks daily attendance.
- Manager – views and manages attendance of all employees.

2. Tech Stack

Frontend: React + Redux Toolkit

Backend: Node.js + Express

Database: MongoDB (Mongoose)

Authentication: JWT

3. Core Features

Employee:

- Register/Login
- Check-In / Check-Out
- View attendance history
- View monthly summary
- Dashboard with stats

Manager:

- Login
- View all employees' attendance
- Filter by employee/date/status
- Export CSV reports
- Team summary & dashboard

4. Database Schema

Users:

id, name, email, password (hashed), role, employeeId, department, createdAt

Attendance:

id, userId, date, checkInTime, checkOutTime, status, totalHours, createdAt

5. Backend API Structure

Auth APIs:

POST /api/auth/register

POST /api/auth/login

GET /api/auth/me

Employee Attendance APIs:

POST /api/attendance/checkin

POST /api/attendance/checkout

GET /api/attendance/my-history

GET /api/attendance/my-summary

GET /api/attendance/today

Manager APIs:

GET /api/manager/all

GET /api/manager/employee/:id

GET /api/manager/summary

GET /api/manager/export

GET /api/manager/today-status

6. Application Flow

Login:

- User logs in → backend returns JWT → stored in localStorage.

Attendance:

- CheckIn saved with time + status (present/late)
- CheckOut saved with total working hours calculation

Manager:

- Views complete team attendance
- Downloads CSV
- Sees present/absent status for the day

7. Frontend Flow

React components call API using Axios.

Redux Toolkit stores:

- Authentication state
- Attendance data
- Manager reports

8. Conclusion

This project demonstrates complete full-stack development including:

- Authentication
- Role-based access
- Real-time attendance management
- Reporting and filtering