Employee Leave Management System

Tech Stack

- Frontend: React + Redux Toolkit/Zustand
- Backend: Node.js + Express
- Database: MongoDB or PostgreSQL

What to Build

A leave management system with 2 user roles:

1. Employee - Apply for leaves
2. Manager - Approve/Reject leaves

Features

Employee Features

1. Register/Login
2. Apply for leave (Sick, Casual, Vacation)
3. View my leave requests with status
4. View leave balance (Sick: 10, Casual: 5, Vacation: 5)
5. Cancel pending requests
6. Dashboard with stats

Manager Features

1. Login
2. View all pending leave requests
3. Approve or Reject leaves
4. View all employees' leave history
5. Dashboard with team stats

Required Pages

Employee:

- Login/Register
- Dashboard
- Apply Leave
- My Requests

- ● Profile

Manager:

- ● Login
- ● Dashboard
- ● Pending Requests
- ● All Requests

Database Schema

Users:

```
- id
- name
- email
- password (hashed)
- role (employee/manager)
- leaveBalance: { sickLeave: 10, casualLeave: 5, vacation: 5 }
```

LeaveRequests:

```
- id
- userId
- leaveType (sick/casual/vacation)
- startDate
- endDate
- totalDays
- reason
- status (pending/approved/rejected)
- managerComment
- createdAt
```

API Endpoints

Auth

```
POST /api/auth/register
POST /api/auth/login
GET  /api/auth/me
```

Leaves (Employee)

```
POST   /api/leaves                - Apply leave
GET    /api/leaves/my-requests    - My requests
DELETE /api/leaves/:id            - Cancel request
GET    /api/leaves/balance        - My balance
```

Leaves (Manager)

```
GET /api/leaves/all          - All requests
GET /api/leaves/pending      - Pending only
PUT /api/leaves/:id/approve  - Approve
PUT /api/leaves/:id/reject   - Reject
```

Dashboard

```
GET /api/dashboard/employee  - Employee stats
GET /api/dashboard/manager   - Manager stats
```

Deliverables

1. GitHub Repository with clean code
2. README.md with:
   - Setup instructions
   - How to run
   - Environment variables
   - Screenshots
3. .env.example file
4. Working application

Evaluation (100 Points)

- Functionality: 40 points
- Code Quality: 25 points
- UI/UX: 15 points
- API Design: 10 points
- Database: 5 points
- Documentation: 5 points

Points are awarded based on your explanation

Submission

- Submit: GitHub repository link & preview url
- Record a video 2-5 minutes explaining the project & share the recorded link.

Good luck!