MERN Internship Project - "QuirkyRoomie" - Flatmate Conflict Management System

---

Project Overview

Welcome to the **MERN Stack Internship Challenge**! This project will test your **end-to-end development skills** using **MongoDB, Express, React, and Node.js**. Your task is to build **QuirkyRoomie**, a web platform that helps flatmates resolve everyday conflicts through complaint logging, voting, and punishments.

Key Skills Being Tested:

✅ Full-Stack MERN Development
✅ Authentication & Authorization (JWT)
✅ MongoDB Schema Design
✅ REST API Development
✅ React Frontend (State Management, Hooks, Routing)
✅ GitHub Repository Management
✅ (Bonus) Deployment & CI/CD

Project Requirements

1. User Authentication & Management (JWT + MongoDB)

- Users can **sign up/login** as flatmates.
- Each user is part of a **flat group** (identified by a **Flat Code**).
- **JWT authentication** required for secured API access.

2. Complaint Logging System (Express + MongoDB)

- Flatmates can **file complaints** about household issues.
- Each complaint should have:
  - **Title**
  - **Description**
  - **Complaint Type** (Noise, Cleanliness, Bills, Pets, etc.)
  - **Severity Level** (Mild, Annoying, Major, Nuclear)

    - **Timestamp**

3. Voting & Community Resolution (React + Express + MongoDB)

- Other flatmates can **upvote/downvote complaints**.
- Complaint with the most upvotes becomes **"Flatmate Problem of the Week."**
- Downvoted complaints **auto-archive after 3 days**.

4. Resolution & Gamification

- Users can **mark complaints as resolved**.
- Flatmates earn **karma points** for resolving issues.
- Monthly **“Best Flatmate” badge** is awarded based on karma scores.

5. Punishment Generator

- If a complaint gets **10+ upvotes**, an automated **punishment** is suggested:
    - "Didn’t clean the dishes? You’re making chai for everyone for a week." ☕
    - "Blasted loud music at 2 AM? You owe everyone samosas." 🥟

6. Leaderboard & Stats

- Displays **who has the most complaints filed against them**.
- Shows **top complaint categories** for the flat.

Tech Stack & Setup Instructions

1. Backend: Node.js + Express

- Use **Express.js** for API development.
- Implement **JWT authentication** for secure user sessions.
- Use **Mongoose** for MongoDB database interaction.

2. Database: MongoDB (Atlas or Local)

- Store users, complaints, votes, and karma points.
- Define relationships (One-to-Many: Flat → Users, Users → Complaints).

3. Frontend: React (Vite or CRA)

- Implement **React Router** for navigation.
- Use **Redux/Context API** for state management.
- Display **leaderboards, complaint tracking, and karma scores**.

4. API Endpoints (Examples)

User Authentication

- POST /api/auth/register – Register new flatmate
- POST /api/auth/login – Login & receive JWT token

Complaint Handling

- POST /api/complaints – File a new complaint (authenticated users only)
- GET /api/complaints – List all active complaints
- PUT /api/complaints/{id}/resolve – Mark complaint as resolved

Voting System

- POST /api/complaints/{id}/vote – Upvote or downvote a complaint
- GET /api/complaints/trending – Fetch most upvoted complaints

Leaderboard & Stats

- GET /api/leaderboard – Show karma points ranking
- GET /api/flat/stats – Get complaint statistics

Project Submission Guidelines

✅ **Push your project to GitHub** (make it public).
✅ Include a **README.md** with setup instructions.
✅ Provide API documentation (Swagger/Postman Collection).
✅ Deploy using **Vercel/Heroku for frontend & Render for backend** (Bonus).
✅ Submit a **5-minute video demo** showcasing your work.

Evaluation Criteria

Category	**Evaluation Criteria**
Backend Development	Well-structured Express.js API, authentication
Security & Authentication	Proper JWT implementation, secure endpoints
Database Handling	Efficient MongoDB schema & queries
Frontend Implementation	Clean React UI, state management
API Documentation	Clear API docs using Swagger/Postman
Creativity & Features	Implementation of punishments, leaderboard, UX
GitHub Activity	Clean commit history, meaningful commits
(Bonus) Deployment	Extra points for Heroku/Vercel/Render deployment

---

Final Words

This project is designed to be **technically challenging**. We are looking for **clean code, structured development, and creativity**. You may not build entire project but efforts will be counted.

Have building, and may the **Best Flatmate Resolver win!** 🚀