MERN Stack Interview Assignment: Multi-Tenant Task Management Platform

hi boo

zxcvb

Overview

i am good how are you

You are required to build a **Multi-Tenant Task Management Platform** using the **MERN Stack** (MongoDB, Express.js, React, Node.js). The project will evaluate your full-stack development skills, understanding of multi-tenancy, backend job scheduling, and containerization/deployment practices.

Note: Feature 1 is mandatory but Feature 2 is good to have

Core Features

1. Multi-Tenant User Management

- **Organization-based Multi-Tenancy**:
  Ensure that all data is isolated per organization. A user should only have access to data belonging to their organization.

- **Role-Based Authentication**:
  Implement 3 roles within an organization:

  - **Admin**: Full access to organization and user management

  - **Manager**: Can manage tasks and assign them to members

  - **Member**: Can view and update assigned tasks

- **User Registration & Onboarding**:

  - Option to **create a new organization** during registration

  - Option to **join an existing organization** via an invite

- **Invitation System**:
  Admins/Managers should be able to invite users to their organization via email or invite link.

- **Authentication**:

    - Use **JWT** for secure session handling
    - Include **role-based access control (RBAC)** at the API level

2. Task Management with Expiry

- **CRUD Operations**:
    - Create, read, update, and delete tasks scoped to the organization
    - Only authorized users can perform these actions based on their roles
- **Task Assignment**:
    - Tasks should be assignable to members within the organization
- **Additional Task Features**:
    - Task **categories** (e.g., Bug, Feature, Improvement)
    - Task **priority** (e.g., Low, Medium, High)
    - **Due date** for each task
- **Automated Task Status & Expiry**:
    - Task statuses: `Todo`, `In Progress`, `Completed`, `Expired`
    - Implement a background job (e.g., with `node-cron`) to auto-update task status to `Expired` once the due date passes
- **Overdue Notifications**:
    - Notify assigned users of overdue tasks (in-app notification preferred)

3. Organization Management (BONUS Section)

- **Dashboard**:
    - Overview of task statistics: total tasks, overdue, completed, per category/priority
- **Member Management**:
    - Admins should be able to view, invite, remove users, and assign roles
- **Organization Settings**:

  - Basic preference configuration like name, theme (optional), etc.

Deliverables

1. Application Code

- Complete MERN stack application
- Multi-tenant architecture implementation
- Task expiry functionality with background jobs
- Role-based access control

2. Containerization

- Docker files for all services
- Docker Compose configuration
- Production-ready container setup

3. Deployment Documentation

- **Server setup guide** with prerequisites
- **CI/CD pipeline** configuration
- **Environment variables** documentation
- **Database migration** scripts

4. Testing Flow

- Unit tests for critical functions
- API endpoint testing
- Container health checks