

Senior React & Node.js Full-Stack Interview Task

Task Overview

Your interview task is to build a full-stack application using React (Next.js), Node.js (NestJS), TypeScript, PostgreSQL, and MongoDB. The application should be a Real-Time Collaborative Task Management System where multiple users can create, assign, and track tasks with real-time updates with usage of WebSockets.

Requirements

Backend (NestJS & TypeORM)

1. Authentication & Authorization

- Implement user authentication using JWT.
- Implement role-based access control (Admin, User).

2. Database & Models

- Use PostgreSQL (via TypeORM) for relational data like users and tasks.
- Use MongoDB for real-time events logging.
- Create models for User, Task, and Event Logs.

3. API Endpoints (NestJS)

- **User Management:** Register/Login, Update Profile.
- **Task Management:**
 - Create, Read, Update, Delete (CRUD) tasks.
 - Assign tasks to users.
- **Real-Time WebSocket Events:**
 - Notify users in real-time when tasks are assigned or updated.

4. Testing

- Write unit tests using Jest for critical services (authentication, task management).

Frontend (React & Next.js)

1. UI Development (React + TypeScript)

- Implement a dashboard showing a list of tasks.
- Allow users to create, assign, and update tasks.
- Follow responsive web design rules.
- Real-time status task updates via WebSockets.

2. State Management

- Use React Context API or Redux for state management.
- Implement caching mechanisms (e.g., React Query).

3. Authentication & Authorization

- Implement JWT-based authentication.
- Protect routes based on user roles (Admin/User).

4. Unit Testing

- Write unit tests using Jest + React Testing Library.

Bonus Points

- Implement CI/CD pipeline for deployment.
- Implement Dockerization for both frontend and backend.
- Use Redis for caching frequently accessed data.
- Deploy on Cloudflare Tunnel for secure tunneling.

Submission Guidelines

- Upload your code to a public GitHub repository.
- Include a README.md with setup instructions.
- Deploy the app (e.g., Vercel + Railway) and provide a demo link.
- Write unit tests covering key functionalities.