Mock Exercise (Take-Home) - Full Stack Engineer

Project: Super-Todo SaaS Platform

Context

You are tasked with developing a new SaaS platform called Super-Todo. You need to set up the project boilerplate and implement core functionalities that will serve as the foundation for future development.

You can use any tools or technologies necessary to meet the following requirements, including AI.

Project Requirements

Technical Stack

- Backend: NestJS
- Frontend: VueJS
- Database: In-memory (for now)
- Repository: Private GitHub repository (Required)
  - Repository name must not contain any reference to the company name
  - Provide access to: `chiquyet199`

User Stories

US1: View To-do List

Display all to-do items in an accordion layout to provide users with an overview of all tasks and their details.

Acceptance Criteria:

- To-do items are displayed in an accordion format
- Each to-do item shows:
  - Title
  - Completion status
  - Creation date
  - Notes section (if the user has a paid plan)
- Clicking on an accordion item expands or collapses it to show or hide details.
- The list automatically updates when items are modified.

US2: Plan-based Features

The system should offer different features based on the user's plan, allowing paid users to access premium features.

Acceptance Criteria:

- System recognizes two types of users via URL parameter: `/?userRole=free` or `/?userRole=paid`
- Free plan users can see:
  - Basic todo information (title, status, dates)
  - CRUD operation with a TodoItem
- Paid plan users can have additionally functionality:
  - Able to add notes for each todo

Technical Requirements

Backend (NestJS)

- Use TypeScript
- Implement in-memory database service
- Tests

Frontend (VueJS)

- Basic styling (Tailwindcss)
- Use TypeScript
- Tests

Deliverables

1. **GitHub Repository** containing:
    - Complete source code
    - README.md with:
        - Setup instructions
        - Architecture decisions
        - Any assumptions made, notes
2. **Working Application**
    - Deployed version of the application (provide URL)
    - Should be functional and demonstrate all required features

Time Expectation

- Estimated Effort: 3-5 hours
- Deadline: 3 calendar days

Submission Process

1. Create a private GitHub repository
2. Add `chiquyet11` as collaborator
3. Deploy application
4. Once you have completed the exercise, share the following URLs to the HR:
    - GitHub repository URL
    - Deployed application URL

Important Reminder

This exercise is designed to evaluate not just your coding abilities, but also your approach to building scalable applications, documentation, and technical planning.