Principal Frontend Engineer Task

Design and Document a Modular Micro Frontend Architecture

Objective:

To assess your fit for the Principal Engineer role at Plan A, we invite you to design a micro frontend architecture that supports multiple teams—reflecting the architecture we currently utilise.
This task will allow you to demonstrate your expertise in architecture design, your ability to create clear and concise documentation, and your skills in mentoring and effective communication.

Task Description:

You are tasked with designing a basic micro frontend architecture for a fictional project, focusing on how it supports independent team development. Your primary goal is to create an architecture that is easy to understand, adopt, and scale, along with clear documentation and a brief outline of mentoring and communication strategies.

Time Expectation:

This task is designed to take less than 4 hours, allowing you to focus on demonstrating your technical skills, architectural thinking, documentation skills, and leadership potential without requiring a full implementation.

planA

Requirements:

Micro Frontend Setup:

- Design a basic architecture that supports multiple teams working on independent micro frontends.
- Create **two** simple micro frontends:
  - **Team Onboarding**:
    - A navigation (This team owns things like login, registration, navigation and data uploads.)
  - **Team Dashboard**:
    - A small dashboard showing one chart using this API. (This team owns the visualisation of the data)
- Implement **one** minimal container application that integrates these micro frontends on one web page.
- Use **TypeScript** for all components and manage dependencies with **Yarn**. Make sure to focus on clean setup and stable dependency management.
- **Bonus**: Automation of tasks like linting or deployments

Documentation:

- **Architecture Overview**: A brief document with at least **one diagram** explaining the architecture and how it supports team autonomy and scalability.
- **Testing**: document your testing strategy.
- **Developer Guide**: A concise guide for developers on how to get started with the architecture, including setup, adding/modifying micro frontends, and handling dependencies.
- **Mentoring and Onboarding**: A short section on how you would mentor developers and onboard new team members to work within this architecture.
- **Stakeholder Communication**: A brief outline on how you would present the architecture to non-technical stakeholders, focusing on its benefits to the business.

Testing:

- Write 1 or 2 basic unit tests for key components using a framework like Jest.

Deliverables:

- ☐ GitHub repository/repositories containing:
  - ☐ The **container application** and **2** micro frontends.
  - ☐ **Documentation** (Architecture Overview with **1 diagram**, Testing, Developer Guide, Mentoring and Onboarding, and Stakeholder Communication).
  - ☐ Basic unit **tests**.
- ☐ A **README** file with a summary and setup instructions.

Technical Interview Follow Up:

- We will talk about the whole task in the interview part, but will focus on the documentation more, especially on Mentoring and Stakeholder Communication.

Evaluation Criteria:

- ☐ **Architectural Design**: Ability to design a simple, scalable, and team-friendly (easy to use, good documented) architecture.
- ☐ **Documentation**: Clarity and conciseness of the provided documentation.
- ☐ **Mentorship and Communication**: Effectiveness in outlining mentoring strategies and communicating with stakeholders.
- ☐ **Technical Implementation**: Proficiency in TypeScript, Yarn, dependency management, and micro frontend integration. **Bonus**: Automation for linters/deployments or ideas on how this could look.
- ☐ **Code Quality**: Clean, maintainable code with adherence to best practices.
- ☐ **Testing**: Adequate coverage and quality in basic testing.