Full Stack Pre-Assessment

Please answer the following question take time no more than 3 working days.

Technology stack

- NestJS
- React
- Postgres

Instruction

- Please create Github public repository to answer the questions below.
- Please reply in 3 days.

Back-end Questions

1. Assuming the system currently has three microservices: Customer API, Master Data API, and Transaction Data API, there is a new feature that requires data from all three microservices to be displayed in near real-time. The current technology stack includes REST APIs and an RDBMS database. How would you design a new API for this feature?
2. Assuming the team has started planning a new project, the project manager asks you for a performance test strategy plan for this release. How would you recommend proceeding to the project manager?
3. Design and develop two APIs using NestJS and Postgres with the following specifications:
    1. **Create a Multilingual Product API**: Develop an API that allows for the creation of products, each with attributes for name and description that support multiple languages.
    2. **Multilingual Product Search API**: Implement an API that enables searching for products by name in any language and returns results in a paginated format.

Additional Requirements:

- **Validation**: Outline how you will validate data inputs in both APIs to ensure data integrity.

- **Database Design**: Describe the database schema and the approach you will use to handle multilingual support for product information.
- **Testing Strategy**: Explain your strategy for testing these APIs, including how you will handle unit tests, integration tests, and any end-to-end testing considerations.

Please provide a detailed explanation of your design decisions for each of these aspects.

React Questions

1. useCallback ใช้ทำอะไร
2. Write a unit test for the UserProfile React component using Jest and React Testing Library.

```
import React, { useState, useEffect } from 'react';

const UserProfile = ({ userId }) => {
  const [user, setUser] = useState(null);
  const [error, setError] = useState('');

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await
fetch(`https://api.example.com/users/${userId}`);
        if (!response.ok) {
          throw new Error('Failed to fetch user data');
        }
        const userData = await response.json();
        setUser(userData);
      } catch (err) {
        setError(err.message);
      }
    };

    fetchData();
  }, [userId]);

  if (error) {
    return <div>Error: {error}</div>;
  }

  if (!user) {
    return <div>Loading...</div>;
  }

  return (
    <div>
      <h1>{user.name}</h1>
      <p>Email: {user.email}</p>
    </div>
  );
};
```

```
export default UserProfile;
```