Front End

Challenge Ssr

Creación de una Aplicación Web con Login y Roles

El objetivo de este desafío es que implementes una aplicación web con un sistema de autenticación básico que permita a dos tipos de usuarios diferentes, "admin" y "user", acceder a información obtenida desde una API que permita operaciones CRUD (POST y PUT). El usuario con rol "user" tendrá acceso de solo lectura a los datos, mientras que el usuario con rol "admin" podrá realizar operaciones de creación y edición.

1

Diseño de interfaz de usuario

Diseña una interfaz de usuario atractiva, accesible e intuitiva. Crea una página de inicio de sesión donde los usuarios puedan ingresar sus credenciales. Crea una página de visualización de información donde se muestren los datos obtenidos desde la API para el usuario "user". Para el usuario "admin", añade opciones de creación y edición de datos.

2

Autenticación y Roles

Implementa un sistema de autenticación básico para validar las credenciales del usuario al iniciar sesión. Define dos roles de usuario: "admin" y "user".
Añade validaciones básicas de formularios, como asegurarse de que los campos no estén vacíos y que las contraseñas tengan una longitud mínima.
Utiliza un contexto o un estado global para manejar roles y permisos, mostrando/ocultando elementos de la interfaz según el rol.
El acceso a creación y edición debera ser por medio de enrutamiento, por lo que se requiere protección de rutas

3

Operaciones CRUD

Utiliza una API que admita operaciones CRUD (POST y PUT) para simular la diferencia de roles. Para el usuario "user", restringe las funcionalidades de edición y crear, mientras que para el usuario "admin", permite realizar todas las operaciones CRUD (excepto DELETE).

Front End

Challenge Ssr

4 Comunicación con la API

Realiza las solicitudes HTTP adecuadas a la API para obtener, crear y actualizar datos. Puedes utilizar bibliotecas como Axios para facilitar las peticiones.

5 Estilos & Maquetación

Utiliza CSS y/o frameworks para implementar el diseño y la estilización de la interfaz de usuario. Organiza los estilos con la estructura de carpetas adecuada y reutilizable. Asegurate que el maquetado sea repsonsivo.
Frameworks sugeridos: Tailwind, Ionic o Material UI

6 Funcionalidades adicionales optativas

- Si lo deseas, puedes agregar funcionalidades adicionales como paginación, filtrado o búsqueda para mejorar la experiencia del usuario.
- TypeScript
- Añade notificaciones para operaciones exitosas o fallidas
- Implementa autenticación con tokens JWT simulados para hacer el desafío más realista.

Requisitos técnicos:

- Utiliza React js, Angular o Next js según lo pautado en la entrevista. Arquitectura de carpetas bien estructurada y organizada para tu proyecto.
- Podrás realizar la entrega de tu trabajo en un repositorio, no es necesario ejecutar el deploy.
- README detallado que fundamente la esctuctura de carpetas, explique cómo instalar, ejecutar y probar la aplicación.
- Api sugerida: https://jsonplaceholder.typicode.com/guide/

Quedamos a tu disposición para cualquier consulta

¡Gracias por tu trabajo!

ELDAR soluciones transaccionales