Teste Técnico: Desenvolvedor Frontend

Introdução

Obrigado pelo seu interesse em se juntar à nossa equipe de desenvolvimento! Este teste técnico foi elaborado para avaliar suas habilidades como desenvolvedor frontend, com foco em Next.js, Next Auth 4 e conhecimentos básicos de backend.

Nossa empresa é uma plataforma de assinatura digital de documentos, então o desafio está alinhado com nosso contexto de negócio.

Objetivo

Desenvolver um protótipo simplificado de uma aplicação de gerenciamento e assinatura de documentos que permita:

1. Autenticação de usuários
2. Upload e listagem de documentos
3. Visualização de documentos
4. Simulação de assinatura digital

Requisitos Técnicos

Frontend

- Utilizar Next.js com App Router
- Implementar autenticação com Next Auth 4
    - Permitir login com e-mail/senha
    - Implementar ao menos um provedor OAuth (Google, GitHub, etc.)
- Criar interfaces responsivas e acessíveis
- Utilizar TypeScript para type safety
- Implementar formulários com validação

Backend (conhecimentos básicos)

- Criar rotas de API no Next.js para gerenciar documentos
- Implementar persistência básica de dados
    - Você pode usar um ORM simples, SQLite, ou até mesmo JSON local
    - Bônus: Implementar com Prisma ORM
- Configurar upload de arquivos (PDF)

Funcionalidades Esperadas

1. **Autenticação**

    - Página de login/registro
    - Proteção de rotas privadas
    - Logout
    - Gerenciamento básico de sessão

2. **Gerenciamento de Documentos**

    - Listagem de documentos do usuário logado
    - Upload de novos documentos (PDF)
    - Visualização de documento
    - Exclusão de documentos

3. **Assinatura Digital (Simplificada)**

    - Interface para simular assinatura em documento
    - Registro da assinatura com timestamp
    - Status do documento (Pendente, Assinado)

Diferenciais

- Testes unitários e/ou de integração
- UI/UX de qualidade (pode utilizar bibliotecas como Tailwind, shadcn/ui, Chakra UI, Material UI, etc.)
- Implementação de contextos e/ou hooks personalizados
- Estratégias de cache e otimização
- Gerenciamento de estado global (Zustand, Redux, Context API, etc.)
- Documentação do código e da aplicação

Estrutura de Dados Sugerida

Users

```
{
  id: string
  name: string
  email: string
  password: string (hashed)
  createdAt: Date
  updatedAt: Date
}
```

Documents

```
{
  id: string
  name: string
  fileKey: string
  userId: string
  status: enum ['PENDING', 'SIGNED']
  createdAt: Date
  updatedAt: Date
}
```

Signatures

```
{
  id: string
  documentId: string
  userId: string
  signatureImg: string (base64 ou URL para imagem)
  createdAt: Date
  signedAt: Date
}
```

Entrega

1. Repositório Git (GitHub, GitLab, Bitbucket) com o código-fonte
2. README.md com:
    - Instruções de instalação e execução
    - Explicação da arquitetura e decisões técnicas
    - Funcionalidades implementadas
    - Tecnologias utilizadas
    - Desafios enfrentados e soluções adotadas
3. Deploy da aplicação (opcional, mas valorizado)
    - Vercel, Netlify, ou qualquer outra plataforma de sua escolha

Prazo

O teste deve ser entregue em até 3 dias após o recebimento.

Critérios de Avaliação

- Qualidade e organização do código
- Cumprimento dos requisitos
- Funcionamento correto das funcionalidades
- Estrutura e arquitetura da aplicação
- Tratamento de erros e edge cases
- Usabilidade e experiência do usuário
- Documentação

Dúvidas

Se tiver qualquer dúvida durante o processo, não hesite em entrar em contato através do e-mail luciano@supersign.com.br.

Boa sorte! Estamos ansiosos para ver sua solução!