Techtins Consultoria Júnior

Documento de Requisitos: Sistema de Gerenciamento de Usuários para E-commerce de Tênis

1. Introdução

A empresa ***SneakerShop***, um e-commerce especializado na venda de tênis de diversas marcas, está em expansão e precisa de uma ferramenta interna para gerenciar os usuários que acessam o sistema, tanto clientes quanto funcionários. A administração da empresa requer um sistema para cadastro, visualização, edição e exclusão de usuários, com um histórico que permite acompanhar as últimas atividades de gerenciamento de usuários.

2. Objetivo do Sistema

O objetivo do sistema é fornecer à ***SneakerShop*** uma aplicação que permita aos administradores gerenciar usuários que fazem parte da operação do e-commerce. O sistema será responsável por realizar o **CRUD de usuários** (Criar, Ler, Atualizar, Deletar), armazenando informações básicas de cada usuário. A interface gráfica deve permitir que os administradores tenham acesso a uma lista de usuários e suas informações, além de um histórico recente das operações realizadas no sistema.

3. Requisitos Funcionais

3.1 Funcionalidades para o Cliente (Admin)

1. O cliente deve ser capaz de:
   - Visualizar uma lista de todos os usuários cadastrados.
   - Cadastrar novos usuários no sistema.
   - Editar informações de usuários existentes.
   - Excluir usuários do sistema.
   - **EXTRA**: Visualizar um histórico dos últimos 5 usuários cadastrados, incluindo informações sobre quem fez o cadastro e quando.

3.2 Informações de cada Usuário

O sistema deve gerenciar as seguintes informações mínimas de cada usuário:

1. Nome Completo.
2. Email.
3. Função (Admin, Cliente, Funcionário).
4. Data de Cadastro.
5. Status (Ativo/Inativo).

3.3 Autenticação e Controle de Acesso

1. O sistema deve permitir que os administradores façam login no sistema de forma segura, utilizando email e senha.
2. Apenas administradores logados podem acessar o sistema e gerenciar os usuários.
3. A segurança das informações deve ser garantida, com todas as rotas protegidas e criptografia de senhas.

4. Requisitos Não Funcionais

1. Segurança: O sistema deve garantir a criptografia das senhas dos usuários e proteger todas as rotas sensíveis com autenticação via token JWT.
3. Manutenção: O código deve ser limpo, modularizado e seguir boas práticas de programação.
4. Usabilidade: A interface deve ser simples, intuitiva e responsiva, permitindo fácil uso em dispositivos móveis e desktops.
5. Documentação: O código e as funcionalidades devem ser documentados no GitHub, com instruções claras sobre instalação e execução de cada método

5. Divisão das Tarefas por Stack

5.1 Backend - Sistema de Segurança (Dois Devs)

- Implementar a lógica de autenticação utilizando JWT.
- Criar as rotas para o CRUD de usuários (criar, listar, editar, deletar) com proteção de acesso.
- Gerenciar permissões de usuários (diferenciar admins e usuários comuns).
- Criar um banco de dados relacional ou não relacional, (o que preferir), para armazenar informações dos usuários.
- Garantir a segurança das rotas com middleware de autenticação.

- Implementar um sistema de logging para registrar as operações feitas no sistema (cadastramento, edição e exclusão de usuários).
- **EXTRA**: Desenvolver a funcionalidade de histórico de usuários recentes, armazenando quem fez o cadastro e a data.
- Criar testes unitários e de integração para garantir o funcionamento adequado da API.

COMUNICAÇÃO ENTRE EQUIPE DE DEVS

- Documentar todas as rotas e funcionalidades da API para facilitar a integração.

5.2 Frontend - Interface Gráfica (Um Dev)

- Desenvolver a interface utilizando algum framework de desenvolvimento, (escolha o que preferir), para criar um sistema visualmente intuitivo.
- Implementar uma página de login para que os administradores possam acessar o sistema de forma segura.
- Desenvolver as páginas de:
  - Lista de Usuários: Exibir todos os usuários cadastrados com a opção de edição e exclusão.
  - Cadastro de Usuário: Formulário para cadastrar novos usuários.
  - **EXTRA**: Histórico de Usuários: Exibir os últimos 5 usuários cadastrados com informações sobre quem os registrou.
- Garantir que a interface seja responsiva e de fácil navegação, com feedback visual adequado (mensagens de sucesso e erro).
- Implementar a comunicação com a API de backend para realizar as operações CRUD e exibir o histórico. **COMUNICAÇÃO ENTRE EQUIPE DE DEVS**.

6. Entrega e Documentação

1. Cada desenvolvedor deve postar sua respectiva parte no GitHub, com um README contendo:
   - Descrição do sistema.
   - Instruções para rodar a aplicação localmente (frontend e backend).
   - Descrição detalhada das rotas e funcionalidades implementadas.
   - Exemplos de como testar as funcionalidades, como preferir, (documentação, texto, vídeo, áudio, desenho, etc).

2. Backend:
   - O backend deve incluir uma documentação detalhada das rotas e da lógica utilizada.
   - As rotas de CRUD e autenticação devem ser descritas com exemplos de chamadas.

3. Frontend:
   - O frontend deve incluir instruções de como configurar o ambiente e conectar com o backend.
   - Descrever a estrutura de pastas e o processo de desenvolvimento da interface, e conexão entre as partes. **COMUNICAÇÃO ENTRE EQUIPE DE DEVS.**

4. Todo o código e documentação devem ser organizados e versionados no GitHub.

7. Considerações Finais

Este projeto tem como foco fornecer uma ferramenta essencial para que a equipe de administração da ***SneakerShop*** gerencie os usuários do e-commerce de forma simples e eficaz. A aplicação deve estar bem documentada e preparada para ser apresentada ao cliente, explicando todas as funcionalidades desenvolvidas e como elas foram integradas para resolver as necessidades da empresa. Ao final, o sistema deve estar funcionando com todas as funcionalidades listadas e rodando localmente para testes e futuras implementações.