Universidade Catalica De Mocambique

Nome : Abdulai Saide Abdala – 706230144

docente: Esmael chande Jr.

Documentação da Aplicação Front-end: Suporte Técnico - Sistema de Atendimento

1. Introdução

Esta documentação descreve todos os aspectos relevantes da aplicação front-end do **Suporte Técnico - Sistema de Atendimento**, uma plataforma web desenvolvida para facilitar o atendimento ao cliente com chat em tempo real, gerenciamento de solicitações e um painel de administração. O objetivo é fornecer uma visão clara das funcionalidades, estrutura, tecnologias utilizadas, instruções de uso e manutenção, bem como informações sobre testes e hospedagem.

Data de Criação: 15/05/2025

Última Atualização: 15/05/2025, 16:34 CAT

2. Informações Gerais

Nome da Aplicação: Suporte Técnico - Sistema de Atendimento

Objetivo Principal: Oferecer uma plataforma para clientes enviarem solicitações de suporte, interagirem com um bot inteligente via chat em tempo real, e escalonarem problemas para administradores, enquanto administradores gerenciam solicitações, usuários, técnicos e mensagens reencaminhadas.

Público-Alvo: Clientes que precisam de suporte técnico e administradores que gerenciam solicitações e equipes técnicas.

Tecnologias Utilizadas:

HTML5, **CSS3**, **JavaScript**: Para estrutura, estilo e interatividade.

Bootstrap 5.3.0: Framework CSS para design responsivo e componentes prontos.

Socket.IO Client: Para comunicação em tempo real via WebSocket.

Fetch API: Para chamadas HTTP ao backend.

Bootstrap Icons 1.10.5: Para ícones visuais.

Google Fonts (Poppins): Para tipografia personalizada.

3. Instalação e Configuração

Pré-requisitos

Node.js: v18 ou superior (recomendado v20 LTS)

npm: Incluído com o Node.js

Sistema Operacional: Windows, macOS ou Linux

Navegador Moderno: Chrome, Firefox ou Edge (para suporte a WebSocket e ES6+)

Comandos de Instalação

Clone o repositório (ou copie os arquivos manualmente):

git clone https://github.com/seu-usuario/suporte-tecnico.git

cd suporte-tecnico

Instale as dependências do projeto (necessárias para o backend, que serve os arquivos front-end):

npm install

Isso instalará express, socket.io, sqlite3 e uuid, que são usados pelo backend para comunicação com o front-end.

Como Iniciar o Projeto

Inicie o servidor (o backend serve os arquivos estáticos do front-end):

npm start

O servidor será iniciado na porta 3000, e os arquivos front-end estarão acessíveis em http://localhost:3000.

Configuração de Variáveis de Ambiente

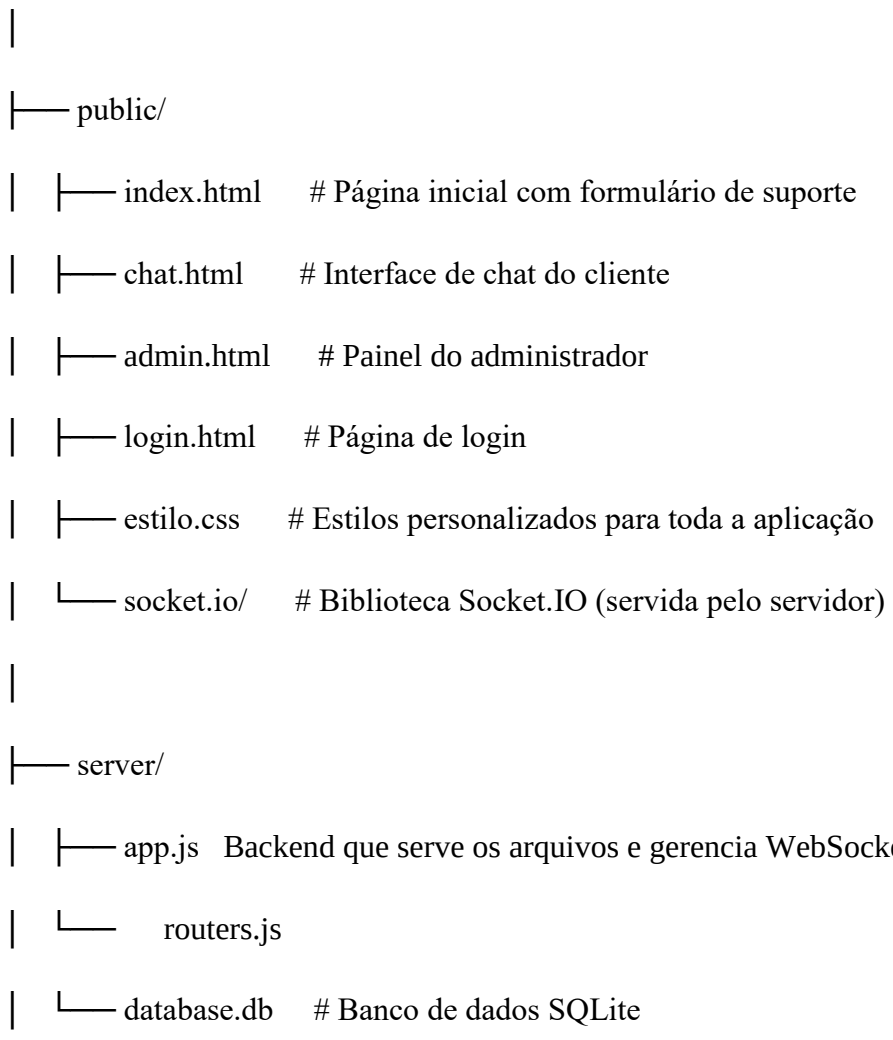
Não há variáveis de ambiente necessárias para o front-end, pois a configuração é gerenciada pelo backend (server/app.js).

Certifique-se de que o backend está configurado para servir os arquivos estáticos da pasta public/.

4. Estrutura do Projeto

A aplicação front-end está localizada na pasta public/. Abaixo está a organização dos arquivos:

suporte-tecnico/



├── package.json   # Dependências e scripts do projeto

└── node_modules/   # Dependências instaladas

Descrição dos Arquivos Front-end

index.html: Página inicial onde os clientes enviam solicitações de suporte com detalhes como nome, produto e problema.

chat.html: Interface de chat em tempo real para clientes, com bot inteligente e reencaminhamento de mensagens após 5 respostas.

admin.html: Painel de administração com seções para gerenciar solicitações, usuários, técnicos, chats e mensagens reencaminhadas.

login.html: Página de login para autenticação de usuários e administradores.

estilo.css: Arquivo CSS com estilos personalizados (ex.: classes modern-card, modern-input) para manter um design consistente.

5. Funcionalidades

Páginas e Funcionalidades Implementadas

Página de Login (login.html):

Permite que usuários e administradores façam login com nome de usuário e senha.

Armazena informações do usuário no sessionStorage para autenticação.

Redireciona para index.html (usuários) ou admin.html (administradores).

Página Inicial (index.html):

Formulário para envio de solicitações de suporte.

Campos: nome, email, idade, sexo, celular, bairro, número da casa, produto e problema.

Após envio, redireciona para chat.html.

Chat em Tempo Real (chat.html):

Interface de chat com bot inteligente que responde a palavras-chave (ex.: "roteador", "computador").

Limite de 5 mensagens por sessão; mensagens adicionais são reencaminhadas ao administrador.

Mensagens são compartilhadas entre abas diferentes do mesmo usuário.

Exibe o nome do usuário logado em suas mensagens e "Bot de Suporte" nas respostas do bot.

Comunicação via WebSocket com o backend.

Painel de Administração (admin.html):

Solicitações Pendentes: Lista solicitações com status "pending".

Problemas Não Resolvidos: Lista solicitações com status "needs_admin".

Dados dos Clientes: Exibe informações detalhadas dos clientes (nome, email, idade, etc.).

Gerenciar Usuários: Permite adicionar, editar e excluir usuários.

Adicionar Técnico: Formulário para adicionar novos técnicos.

Designar Técnico: Permite atribuir solicitações a técnicos e agendar visitas.

Chat com o Cliente: Interface para visualizar e responder a chats de clientes.

Mensagens Reencaminhadas: Seção para visualizar e responder mensagens enviadas após o limite de 5 mensagens.

Responsividade:

Uso do Bootstrap para garantir que a aplicação seja responsiva em dispositivos móveis e desktops.

Layouts ajustados com classes como col-md-6 para formulários e tabelas.

Integração com Backend:

Chamadas HTTP (fetch) para endpoints como /api/requests, /api/users, e /api/technicians.

Comunicação em tempo real via WebSocket (socket.io) para chat e notificações.

6. Documentação de Código

Comentários no Código:

O código JavaScript em cada arquivo HTML contém comentários descritivos para cada função principal (ex.: // Carregar solicitações pendentes em admin.html).

Seções críticas, como a lógica do limite de mensagens em chat.html, são comentadas para facilitar manutenção.

Boas Práticas Adotadas:

Uso de async/await para chamadas assíncronas (ex.: fetch para carregar solicitações).

Manipulação segura de erros com blocos try/catch.

Código modularizado com funções específicas (ex.: loadRequests(), loadTechnicians() em admin.html).

Estilização consistente com classes CSS reutilizáveis (ex.: modern-card, modern-input).

Ferramentas de Documentação:

Não foram utilizadas ferramentas como JSDoc, mas o código está organizado e comentado para fácil entendimento.

Recomenda-se adicionar JSDoc para documentação formal em futuras iterações.

7. Testes

Tipos de Testes

Testes Manuais:

Testes de funcionalidade em chat.html para verificar o limite de 5 mensagens e reencaminhamento.

Testes de responsividade em diferentes dispositivos (mobile, tablet, desktop).

Testes de integração com o backend (ex.: envio de solicitações, carregamento de usuários).

Testes de Integração:

Verificação da comunicação WebSocket entre chat.html e admin.html (ex.: mensagens reencaminhadas aparecem corretamente).

Teste de chamadas HTTP para garantir que solicitações e usuários sejam carregados corretamente.

Ferramentas Utilizadas

Não foram implementados testes automatizados.

Recomendação: Adicionar testes unitários e de integração com ferramentas como:

Jest: Para testes unitários de funções JavaScript.

Cypress: Para testes end-to-end, simulando interações do usuário.

Como Executar Testes

Atualmente, os testes são manuais. Para testar:

Inicie o servidor com npm start.

Acesse http://localhost:3000 e siga os passos de uso descritos no README.

Verifique os logs no console do navegador (DevTools) e do servidor para identificar erros.

8. Hospedagem

A aplicação ainda não foi hospedada em uma plataforma pública, mas pode ser implantada em serviços como Netlify, Vercel ou Heroku. Abaixo estão os passos sugeridos para deploy:

Passos para Deploy

Prepare o Projeto:

Certifique-se de que todas as dependências estão listadas no package.json.

Adicione um script de build, se necessário (atualmente, o projeto usa arquivos estáticos).

Hospedagem no Heroku (exemplo):

Instale o Heroku CLI e faça login:

heroku login

Crie uma aplicação Heroku:

heroku create nome-da-sua-aplicacao

Envie o projeto:

git push heroku main

Configure a porta no app.js para usar process.env.PORT.

Configuração de Domínio (se aplicável):

No Heroku, adicione um domínio personalizado via:

heroku domains:add seu-dominio.com

Configure o DNS no provedor do domínio para apontar para o endereço fornecido pelo Heroku.

Link de Acesso:

Após o deploy, acesse a URL fornecida (ex.: https://nome-da-sua-aplicacao.herokuapp.com).

Observação

Atualmente, a aplicação roda localmente em http://localhost:3000.

Recomenda-se hospedar o backend e os arquivos estáticos juntos, pois o front-end depende do WebSocket servido pelo backend.

9. Conclusão

O **Suporte Técnico - Sistema de Atendimento** é uma aplicação front-end funcional e interativa que atende às necessidades de suporte técnico com chat em tempo real e gerenciamento administrativo. Durante o desenvolvimento, foram superados desafios como a sincronização de mensagens entre abas e o reencaminhamento de mensagens para administradores, resultando em uma experiência de usuário fluida e eficiente.

Lições Aprendidas

WebSocket: A integração com Socket.IO foi essencial para comunicação em tempo real, mas exigiu ajustes para lidar com múltiplas abas.

Limite de Mensagens: Implementar o limite de 5 mensagens e o reencaminhamento foi um desafio que exigiu coordenação entre front-end e backend.

Responsividade: O uso do Bootstrap facilitou a criação de layouts responsivos, mas ajustes personalizados em estilo.css foram necessários para consistência visual.

Sugestões para Melhorias Futuras

Testes Automatizados: Implementar testes unitários e end-to-end com Jest e Cypress para aumentar a confiabilidade.

Notificações: Adicionar notificações push ou sonoras para novas mensagens ou solicitações.

Autenticação Avançada: Substituir o sessionStorage por um sistema de autenticação mais seguro, como JWT.

Internacionalização: Adicionar suporte a múltiplos idiomas para atender a um público mais amplo.

Melhorias no Bot: Expandir as capacidades do bot com aprendizado de máquina ou mais respostas predefinidas.

Desenvolvido por: [Abdulai Saide Abdala]
E-mail: kennysaide2003@gmail.com
Data: 15/05/2025, 16:34
Agradecemos por usar o Suporte Técnico! 🚀