

PROMPT DE ENGENHARIA: SISTEMA DE GESTÃO TOTAL (CMS) - MD SOLUTION

Contexto: Atue como um Desenvolvedor Full Stack Senior e Arquiteto de Sistemas. O objetivo é reconstruir/evoluir o painel administrativo do projeto **MD Solution**, transformando-o em um CMS (Content Management System) proprietário, ultra-flexível e conectado ao banco de dados (Supabase/PostgreSQL).

Instrução Principal:

Desenvolva um plano de implementação e o código base para um **Painel Administrativo Full-Control**. O sistema deve permitir a edição granular de cada componente do site, com persistência de dados e controle de estado em tempo real.

1. Especificações Técnicas do Painel (Funcionalidades):

- **Edição por Seções (Toggle On/Off):** Implementar um sistema de visibilidade para cada seção (Hero, Body, Serviços, Provas Sociais, Blog, Rodapé, Cabeçalho). O admin deve poder ativar/desativar páginas ou seções específicas com um clique.
- **Granularidade de Design:**
 - **Tipografia:** Seleção de Font-Family (Google Fonts), Font-Size (slider), Font-Weight e Line-Height para Títulos (H1-H6) e Subtítulos.
 - **Cores e Estilos:** Color picker para cores primárias, secundárias, fundos de seções, cores de botões (hover/active) e bordas.
 - **Botões e Links:** Editor de texto de CTA, destino do link e estilo do botão (outline/filled/rounded).
- **Gestão de Conteúdo Dinâmico:**
 - **Cards de Serviço e Preço:** Manter o layout atual, mas tornar os textos, ícones e valores editáveis via admin.
 - **Provas Sociais:** CRUD para depoimentos e logotipos de parceiros.
 - **Blog:** Interface completa de publicação (Rich Text Editor) com upload de imagens e SEO dinâmico.
- **Filtros por Página:** O admin deve ter um seletor de páginas (Home, Landing Pages, Blog) que filtre apenas os elementos editáveis daquela página específica.

2. Estrutura do Roadmap de Implementação Requerido:

Por favor, gere o plano detalhado seguindo esta estrutura:

1. **Objetivos e Contexto:** Definição do estado atual (estático/semi-estático) para o desejado (painel dinâmico governado por JSON/Banco de Dados).
2. **Requisitos e Restrições:** Uso de React/Next.js, Tailwind CSS, Lucide Icons e integração com o Schema do Banco de Dados existente.
3. **Fases do Roadmap:**
 - *Fase 1 (Arquitetura):* Modelagem do banco de dados (tabelas de configurações e conteúdo).
 - *Fase 2 (Core Admin):* Estrutura de navegação e filtros por página.
 - *Fase 3 (Editor Visual):* Implementação dos controles de tipografia e cores.
 - *Fase 4 (Blog e Mídia):* Sistema de upload e editor de texto.
4. **Processo Passo a Passo:** Descrição técnica de como conectar o Admin ao Front-end (ex: uso de Context API ou variáveis CSS dinâmicas).

5. **Desafios e Soluções:** Como lidar com performance (Cumulative Layout Shift) ao carregar estilos dinâmicos e segurança de acesso.
6. **Conformidade e Melhores Práticas:** LGPD para coleta de leads, acessibilidade (contraste de cores automático) e indicadores de sucesso (tempo de carregamento < 2s).

3. Saída Esperada:

- **Roadmap Textual:** Completo com prazos sugeridos e KPIs.
 - **Schema SQL:** Código para as tabelas `site_settings` , `page_sections` e `blog_posts` .
 - **Protótipo de Código:** Um componente React de exemplo (`AdminSectionEditor`) que demonstre o controle de um Card de Serviço (Título, Cor, Toggle Ativo).
-

Dica para usar no Antigravity:

Ao colar este prompt, se o Antigravity pedir por onde começar, diga: **"Comece gerando o Roadmap detalhado e o Schema do Banco de Dados (SQL), pois precisaremos dessa base para a persistência das edições."**