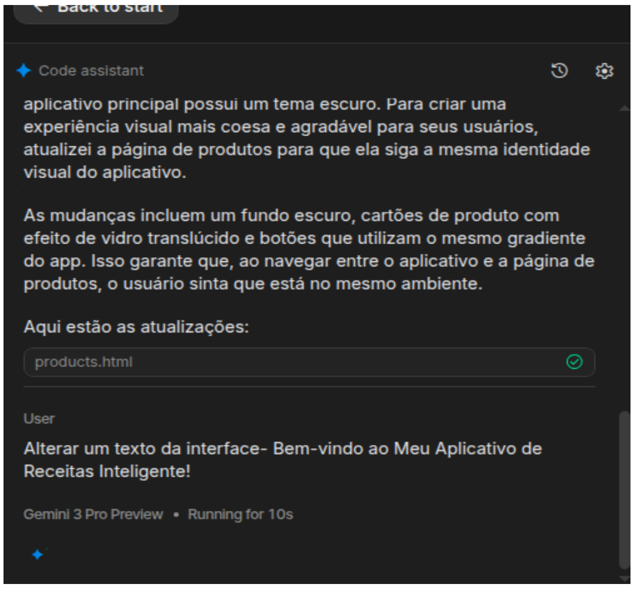
PROMPTS UTILIZADOS PARA A ALTERAÇÃO:



aplicativo principal possui um tema escuro. Para criar uma experiência visual mais coesa e agradável para seus usuários, atualizei a página de produtos para que ela siga a mesma identidade visual do aplicativo.

As mudanças incluem um fundo escuro, cartões de produto com efeito de vidro translúcido e botões que utilizam o mesmo gradiente do app. Isso garante que, ao navegar entre o aplicativo e a página de produtos, o usuário sinta que está no mesmo ambiente.

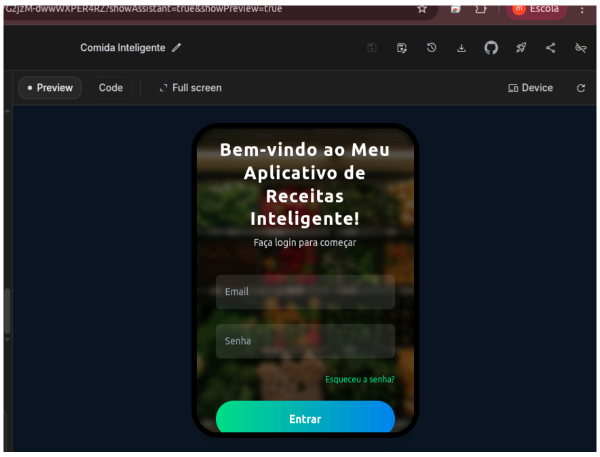
Aqui estão as atualizações:

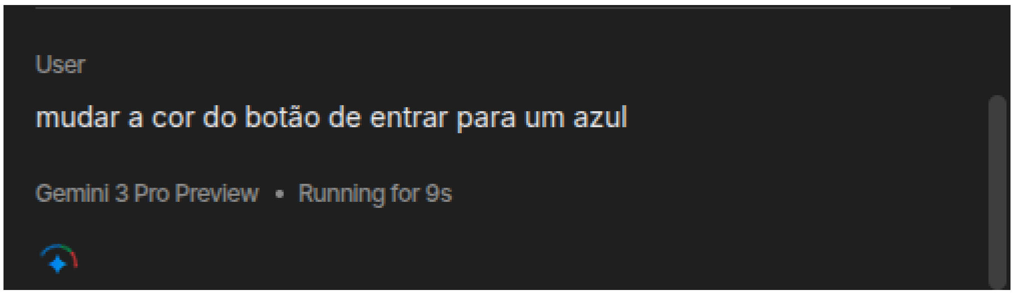
products.html

User

Alterar um texto da interface- Bem-vindo ao Meu Aplicativo de Receitas Inteligente!

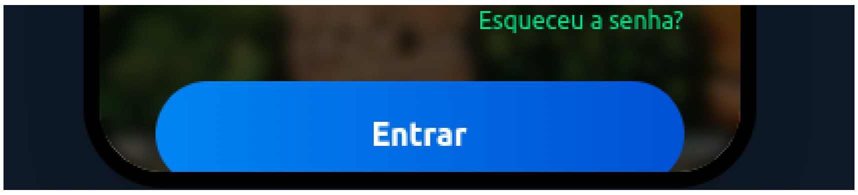
Gemini 3 Pro Preview • Running for 10s

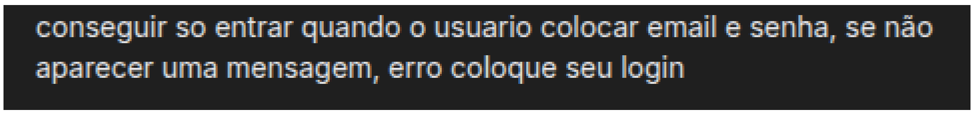



User

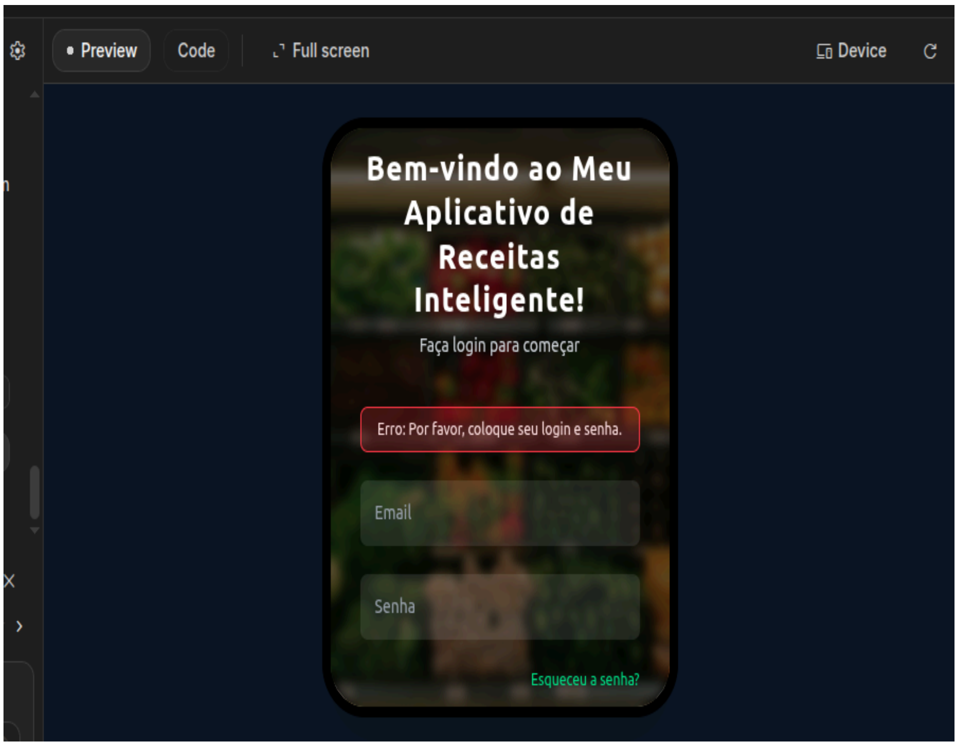
mudar a cor do botão de entrar para um azul

Gemini 3 Pro Preview • Running for 9s





conseguir so entrar quando o usuario colocar email e senha, se não aparecer uma mensagem, erro coloque seu login



Bem-vindo ao Meu Aplicativo de Receitas Inteligente!

Faça login para começar

Erro: Por favor, coloque seu login e senha.

código do arquivo App`.tsx`:

```
import React, { useState, useCallback } from 'react';
import { Header } from './components/Header';
import { IdentifyButton } from './components/IdentifyButton';
import { IngredientList } from './components/IngredientList';
import { RecipeSuggestions } from './components/RecipeSuggestions';
import { BottomNav } from './components/BottomNav';
import type { Recipe, UserProfile } from './types';
import { generateRecipes, identifyIngredientsFromImage } from
'./services/geminiService';
import { RecipeDetailView } from './components/RecipeDetailView';
import { ProfileView } from './components/ProfileView';
import { CameraView } from './components/CameraView';
import { AuthView } from './components/AuthView';
import { FavoritesView } from './components/FavoritesView';
import { ShoppingListView } from './components/ShoppingListView';

const BACKGROUND_URL =
'https://images.unsplash.com/photo-1542838132-92c53300491e?q=80&w=1974&auto=format&
fit=crop';

const App: React.FC = () => {
 const [isAuthenticated, setIsAuthenticated] = useState<boolean>(false);
 const [ingredients, setIngredients] = useState<string[]>([]);
 const [recipes, setRecipes] = useState<Recipe[]>([]);
 const [favoriteRecipes, setFavoriteRecipes] = useState<Recipe[]>([]);
 const [shoppingList, setShoppingList] = useState<string[]>([]);
 const [isLoading, setIsLoading] = useState<boolean>(false);
 const [loadingMessage, setLoadingMessage] = useState<string>('');
 const [error, setError] = useState<string | null>(null);
 const [activeTab, setActiveTab] = useState<string>('Início');
 const [selectedRecipe, setSelectedRecipe] = useState<Recipe | null>(null);
 const [isCameraOpen, setIsCameraOpen] = useState<boolean>(false);
 const [profile, setProfile] = useState<UserProfile>({
   name: '',
   age: '',
   gender: 'nao_informar',
   dietaryRestrictions: [],
 });

 const handleGenerateRecipes = useCallback(async (currentIngredients: string[]) =>
{
   if (currentIngredients.length === 0) {
     setRecipes([]);
     return;
   }

   setLoadingMessage('Sugerindo receitas...');
   setIsLoading(true);
   setError(null);
   try {
     const suggestedRecipes = await generateRecipes(currentIngredients, profile);
     setRecipes(suggestedRecipes);
   } catch (err) {
     setError('Falha ao buscar receitas. Tente novamente mais tarde.');
     console.error(err);
   } finally {
     setIsLoading(false);
     setLoadingMessage('');
   }
 }, [profile]);

 const handleCapture = useCallback(async (imageBase64: string) => {
   setIsCameraOpen(false);
   setIsLoading(true);
   setLoadingMessage('Identificando ingredientes...');
   setError(null);
   setIngredients([]);
   setRecipes([]);
   try {
     const identified = await identifyIngredientsFromImage(imageBase64);
     setIngredients(identified);
     if (identified.length > 0) {
       await handleGenerateRecipes(identified);
     } else {
       setIsLoading(false);
       setLoadingMessage('');
     }
   } catch (err) {
     setError('Falha ao identificar ingredientes. Tente uma foto mais nítida.');
     console.error(err);
     setIsLoading(false);
     setLoadingMessage('');
   }
 }, [handleGenerateRecipes]);

 const handleIdentifyClick = () => {
   if (ingredients.length > 0) {
     handleGenerateRecipes(ingredients);
   } else {
     setIsCameraOpen(true);
   }
 };
  const handleToggleFavorite = (recipe: Recipe) => {
   setFavoriteRecipes(prev => {
     const isFavorite = prev.some(r => r.recipeName === recipe.recipeName);
     if (isFavorite) {
       return prev.filter(r => r.recipeName !== recipe.recipeName);
     } else {
       return [...prev, recipe];
     }
   });
 };

 const isRecipeFavorite = (recipe: Recipe) => {
   return favoriteRecipes.some(r => r.recipeName === recipe.recipeName);
 }

 const handleAddIngredient = (ingredient: string) => {
   if (!ingredients.includes(ingredient)) {
     setIngredients(prev => [...prev, ingredient]);
   }
 };

 const handleRemoveIngredient = (index: number) => {
   setIngredients(prev => prev.filter((_, i) => i !== index));
 };

 const handleProfileSave = (updatedProfile: UserProfile) => {
   setProfile(updatedProfile);
   alert('Perfil salvo com sucesso!');
 };

 const renderContent = () => {
   if (selectedRecipe) {
     return <RecipeDetailView
```

```
       recipe={selectedRecipe}
       onBack={() => setSelectedRecipe(null)}
       onToggleFavorite={handleToggleFavorite}
       isFavorite={isRecipeFavorite(selectedRecipe)}
     />;
   }

   switch (activeTab) {
     case 'Início':
       return (
         <main className="flex-grow overflow-y-auto px-6 py-4 space-y-6">
           <div className="flex justify-center">
             <IdentifyButton
               onClick={handleIdentifyClick}
               loading={isLoading}
               text={ingredients.length > 0 ? 'GERAR NOVAS RECEITAS' : 'IDENTIFICAR
ALIMENTOS'}
               loadingText={loadingMessage}
             />
           </div>
           <IngredientList
             ingredients={ingredients}
             onAddIngredient={handleAddIngredient}
             onRemoveIngredient={handleRemoveIngredient}
           />
           {error && <p className="text-center text-red-400 bg-red-900/50 p-3
rounded-lg">{error}</p>}
           <RecipeSuggestions recipes={recipes} isLoading={isLoading}
onRecipeSelect={setSelectedRecipe} />
         </main>
       );
     case 'Favoritos':
       return <FavoritesView recipes={favoriteRecipes}
onRecipeSelect={setSelectedRecipe} />;
     case 'Lista de Compras':
       return <ShoppingListView items={shoppingList} />;
     case 'Perfil':
       return <ProfileView profile={profile} onSave={handleProfileSave} />;
     default:
       return null;
   }
 };

 const appShell = (content: React.ReactNode) => (
    <div
     className="relative w-full max-w-sm h-[800px] max-h-[90vh] bg-cover bg-center
rounded-[40px] shadow-2xl overflow-hidden border-8 border-black flex flex-col"
     style={{ backgroundImage: `url(${BACKGROUND_URL})` }}
   >
     <div className="absolute inset-0 bg-black/60 backdrop-blur-sm"></div>
     <div className="relative z-10 flex flex-col h-full">
```

```
       {content}
     </div>
   </div>
 )

 if (!isAuthenticated) {
   return (
     <div className="min-h-screen w-full flex items-center justify-center font-sans
p-4 bg-gray-900">
       {appShell(<AuthView onLoginSuccess={() => setIsAuthenticated(true)} />)}
     </div>
   );
 }

 return (
   <div className="min-h-screen w-full flex items-center justify-center font-sans
p-4 bg-gray-900">
     {appShell(
       <>
         {isCameraOpen && <CameraView onCapture={handleCapture} onClose={() =>
setIsCameraOpen(false)} />}
         <Header />
         {renderContent()}
         {!selectedRecipe && <BottomNav activeTab={activeTab}
onTabChange={setActiveTab} />}
       </>
     )}
   </div>
 );
};

export default App;
```