project:

name: "Vibea+"

version: "1.1"

type: "dating\_relationship\_app"

language: "portuguese"

description: |

Aplicativo de relacionamento único com três modos de descoberta flexíveis:

- Deslizar (swipe tradicional)

- Próximos (usuários próximos em tempo real)

- Caminhos (cruzamentos baseados em geolocalização)

Foco em personalização, privacidade e experiência do usuário premium.

# ARQUITETURA TÉCNICA

tech\_stack:

frontend:

framework: "Flutter 3.x"

language: "Dart"

architecture: "Clean Architecture (Domain/Data/Presentation)"

state\_management: "Riverpod ou Bloc"

navigation: "go\_router"

ui\_framework: "Material 3 custom"

maps: "Google Maps SDK"

location: "geolocator + foreground service"

backend:

platform: "Firebase"

services:

- "Firebase Auth"

- "Firestore"

- "Cloud Storage"

- "Cloud Functions (Node 20)"

- "Firebase Cloud Messaging (FCM)"

- "Firebase Analytics"

- "Remote Config"

billing: "RevenueCat"

moderation: "Cloud Vision API"

cdn: "Firebase Storage + resizes"

observability:

- "Crashlytics"

- "Firebase Analytics"

- "Performance Monitoring"

- "Custom dashboards"

# DESIGN SYSTEM

ui\_ux:

theme:

default\_mode: "dark" # Modo escuro como padrão

design\_style: "Neumorfismo sutil"

shadows: "Suaves para efeito extrudado"

borders: "1-2px com opacidade baixa"

dark\_theme:

primary\_color: "#000000" # Preto base

card\_color: "#0B0B0B"

surface\_color: "#121212"

text\_primary: "#FFFFFF"

text\_secondary: "#B3B3B3"

light\_theme:

primary\_color: "#FFFFFF" # Branco base

card\_color: "#F5F5F5"

surface\_color: "#FAFAFA"

text\_primary: "#000000"

text\_secondary: "#666666"

mode\_colors:

dark\_theme:

deslizar: "#1F1F1F" # Cinza-grafite

proximos: "#2C2C2C" # Cinza médio

caminhos: "#7C5CFC" # Vibrante roxo

light\_theme:

deslizar: "#E8E8E8" # Cinza claro

proximos: "#E0E0E0" # Neutro claro

caminhos: "#9B7DFF" # Roxo mais claro

theme\_switching:

user\_controlled: true

system\_follow: true # Seguir configuração do sistema

automatic\_scheduling: false # Opção futura para mudança automática

smooth\_transitions: true # Animações suaves na troca

accessibility:

contrast: "AA compliant em ambos os temas"

fonts: "Escaláveis (Dynamic Type)"

screen\_readers: "VoiceOver/TalkBack support"

touch\_targets: "≥44x44px"

# FUNCIONALIDADES PRINCIPAIS

features:

mvp:

authentication:

- "Login por telefone + SMS"

- "Sign in with Apple"

- "Sign in with Google"

- "Recuperação de senha"

- "Verificação de idade 18+"

profile:

required\_fields:

- "Nome completo"

- "Idade (≥18)"

- "Gênero (obrigatório)"

- "Foto principal"

optional\_fields:

- "4-5 fotos adicionais"

- "Bio/descrição"

- "Altura, tipo físico"

- "Tribo/estilo (LGBTQIA+)"

- "Interesses e hobbies"

- "Integração Instagram/Spotify"

preferences:

- "Gêneros para exibir (multisseleção)"

- "Objetivos (namoro, casual, amizade)"

- "Configurações de privacidade"

discovery\_modes:

deslizar:

description: "Stack de cartões tradicional"

actions: ["like", "dislike", "super\_like"]

limits: "Freemium com limites para usuários gratuitos"

proximos:

description: "Usuários próximos em tempo real"

display: "Lista/grade por distância"

update: "Tempo real com geolocalização"

caminhos:

description: "Cruzamentos baseados em rotas"

tracking: "Contínuo em background"

privacy: "Geohash agregado/privado"

notifications: "Cruzamentos detectados"

messaging:

- "Chat texto e imagens"

- "Moderação automática de conteúdo"

- "Status de leitura opcional"

- "Notificações push"

- "Bloqueio e denúncia"

monetization:

subscription:

plans: ["Mensal", "Trimestral", "Anual"]

features:

- "Ver curtidas recebidas"

- "Filtros avançados"

- "Caminhos sem limite"

- "Boost de perfil"

consumables:

vibes:

- "Super Like premium"

- "Boost de visibilidade"

future\_roadmap:

fase\_2: ["Stories", "Videochamadas", "IA para sugestões"]

fase\_3: ["Eventos ao vivo", "Integrações externas ampliadas"]

fase\_4: ["Sugestões inteligentes por comportamento"]

# MODELAGEM DE DADOS

database:

platform: "Firestore"

collections:

users:

path: "/Users/{userId}"

fields:

- "displayName: string"

- "birthDate: timestamp"

- "gender: enum(man,woman,nonbinary,trans,other)"

- "showGenders: string[]"

- "bio: string"

- "objectives: string[]"

- "photos: {mainUrl, gallery[]}"

- "locVisibility: enum(off,approx,precise)"

- "isPremium: boolean"

- "flags: {isVerified, banned}"

swipes:

path: "/Swipes/{userId}/items/{targetUserId}"

fields:

- "action: enum(like,dislike,superLike)"

- "createdAt: timestamp"

matches:

path: "/Matches/{pairId}"

fields:

- "users: [userIdA, userIdB]"

- "createdAt: timestamp"

- "active: boolean"

conversations:

path: "/Conversations/{threadId}/Messages/{messageId}"

fields:

- "from: userId"

- "type: enum(text,image)"

- "content: string"

- "createdAt: timestamp"

- "readBy: string[]"

geo\_tracking:

path: "/Geo/{userId}/pings/{pingId}"

fields:

- "lat: number"

- "lng: number"

- "accuracy: number"

- "createdAt: timestamp"

crossings:

path: "/Crossings/{userId}/events/{eventId}"

fields:

- "otherUserId: string"

- "at: timestamp"

- "approxAreaHash: string"

security\_rules:

- "Menor privilégio possível"

- "Usuários só acessam próprios dados"

- "Mensagens restritas a participantes"

- "Validação server-side em Cloud Functions"

# CLOUD FUNCTIONS CRÍTICAS

cloud\_functions:

onSwipeWrite:

trigger: "Firestore write em /Swipes"

action: "Verifica match mútuo → Cria Match e Conversation"

onImageUpload:

trigger: "Storage upload"

action: "Moderação + resize + atualização perfil"

onLocationPing:

trigger: "Geolocalização update"

action: "Processa geohash + detecta cruzamentos"

onPurchaseUpdated:

trigger: "RevenueCat webhook"

action: "Atualiza status Premium do usuário"

# GEOLOCALIZAÇÃO E PRIVACIDADE

location:

android:

permissions:

- "ACCESS\_FINE\_LOCATION"

- "ACCESS\_COARSE\_LOCATION"

- "ACCESS\_BACKGROUND\_LOCATION"

- "FOREGROUND\_SERVICE"

- "FOREGROUND\_SERVICE\_LOCATION"

service: "Foreground Service com notificação persistente"

ios:

info\_plist:

- "NSLocationWhenInUseUsageDescription"

- "NSLocationAlwaysAndWhenInUseUsageDescription"

background\_modes: ["Location Updates", "Remote notifications"]

privacy\_strategy:

- "Intervalos adaptativos (5-10min parado, 1-3min movimento)"

- "Debounce por distância (≥50-100m)"

- "Geohash reduzido para privacidade"

- "Controle total pelo usuário"

# SEGURANÇA E MODERAÇÃO

security:

authentication:

- "Verificação telefone + reCAPTCHA"

- "Sign in with Apple/Google"

- "DeviceCheck/SafetyNet"

content\_moderation:

automatic:

- "Cloud Vision API para imagens"

- "Filtros de texto automáticos"

manual:

- "Fila de revisão humana"

- "Sistema de denúncia"

- "Bloqueio por severidade"

data\_protection:

- "Criptografia TLS em trânsito"

- "Dados sensíveis não indexáveis"

- "LGPD/GDPR compliance"

- "Fluxo de exclusão completa"

- "Portabilidade de dados"

# NOTIFICAÇÕES

notifications:

platform: "Firebase Cloud Messaging + APNs"

types:

- "match.created: É um Match!"

- "message.new: Nova mensagem"

- "crossing.detected: Cruzamento detectado"

- "promo: Ofertas promocionais"

features:

- "Deep links para Thread/Perfil/Modo"

- "Silenciamento por conversa"

- "DND por horários"

- "Notificações inteligentes contextuais"

# ANALYTICS E KPIS

analytics:

events:

onboarding: ["signup\_start", "signup\_complete", "profile\_completed"]

engagement: ["mode\_view", "swipe\_like", "swipe\_dislike", "super\_like"]

conversion: ["match\_created", "message\_sent", "crossing\_detected"]

monetization: ["purchase\_started", "purchase\_success", "purchase\_refund"]

kpis:

growth: ["Taxa de cadastro", "Perfil 100% completo"]

engagement: ["Match rate", "Tempo médio sessão", "Modos mais usados"]

retention: ["D1/D7/D30", "Churn rate", "Sessions per user"]

monetization: ["ARPU", "Conversion rate", "LTV"]

# TESTES E QA

testing:

unit\_tests:

- "ViewModels e UseCases"

- "Lógica de matching"

- "Validações de dados"

widget\_tests:

- "Cartões de perfil"

- "Listas e grids"

- "Telas principais"

integration\_tests:

- "Fluxo completo: onboarding → match → chat"

- "Geolocalização e cruzamentos"

- "Compras e assinaturas"

performance\_tests:

- "Carga de usuários simultâneos"

- "Consumo de bateria (modo Caminhos)"

- "Tempo carregamento <2s"

- "Compressão e cache de imagens"

beta\_testing:

ios: "TestFlight"

android: "Google Play Internal Testing"

distribution: "Firebase App Distribution"

# PUBLICAÇÃO NAS LOJAS

app\_stores:

ios\_app\_store:

bundle\_id: "com.vibea.app"

requirements:

- "Privacy Nutrition Labels completos"

- "Sign in with Apple obrigatório"

- "Justificar Background Location"

- "Política privacidade + Termos uso"

certificates: "Push notifications + Background modes"

google\_play:

package: "com.vibea.app"

requirements:

- "Data Safety Section preenchida"

- "Justificar permissões background explicitamente"

- "Play App Signing configurado"

features: "Location + Camera + Storage permissions"

aso\_assets:

- "Screenshots otimizados por modo"

- "Vídeo promocional 30s"

- "Descrição ASO com keywords"

- "Ícone discreto e profissional"

# MARKETING E CRESCIMENTO

marketing:

landing\_page: "Next.js responsiva"

acquisition:

paid: ["Google Ads", "TikTok Ads", "Instagram Ads", "Facebook Ads"]

organic: ["SEO", "Content marketing", "Influencer partnerships"]

retention:

- "Programa indicação com deep link"

- "Push notifications personalizadas"

- "Gamificação (badges, streaks)"

- "A/B testing contínuo"

- "Email marketing segmentado"

growth\_hacks:

- "Onboarding simplificado"

- "Tutorial interativo dos modos"

- "Ofertas limitadas para conversão"

- "Social proof e testimonials"

# CI/CD E DEPLOYMENT

deployment:

ci\_cd: "GitHub Actions + Fastlane"

environments:

development: "Firebase project dev"

staging: "Firebase project staging + beta builds"

production: "Firebase project prod"

pipeline\_stages:

- "Code quality + linting"

- "Unit + widget tests"

- "Build iOS/Android"

- "Integration tests"

- "Deploy staging"

- "Manual QA approval"

- "Deploy production"

- "Store submission"

# REQUISITOS DE COMPLIANCE

compliance:

age\_verification: "18+ obrigatório com validação"

data\_protection:

lgpd: "Lei Geral Proteção Dados (Brasil)"

gdpr: "General Data Protection Regulation (EU)"

content\_policy: "Moderação proativa + reativa"

safety: "Sistema robusto denúncia e bloqueio"

# ESPECIFICAÇÕES DE PERFORMANCE

performance:

targets:

load\_time: "<2s para telas principais"

image\_compression: "Automática com múltiplos tamanhos"

offline\_support: "Conversas e perfis em cache"

battery\_optimization: "Localização adaptativa"

monitoring:

- "Firebase Performance"

- "Crashlytics com alertas"

- "Custom metrics dashboard"

- "Real user monitoring (RUM)"

# ROADMAP DE DESENVOLVIMENTO

timeline:

fase\_1\_mvp: "4-6 meses"

- "Onboarding + perfil completo"

- "3 modos descoberta básicos"

- "Chat + moderação"

- "Monetização freemium"

- "Beta testing"

fase\_2\_enhancement: "2-3 meses"

- "Stories e videochamadas"

- "IA para sugestões"

- "Verificação avançada"

fase\_3\_scaling: "3-4 meses"

- "Eventos ao vivo"

- "Integrações externas"

- "Features sociais"

fase\_4\_intelligence: "4-6 meses"

- "ML para matching"

- "Recomendações comportamentais"

- "Analytics preditivos"

success\_criteria:

technical:

- "App store approval primeira tentativa"

- "Crash rate <1%"

- "Performance targets atingidos"

- "Security audit aprovado"

business:

- "1000+ usuários beta com feedback positivo"

- "10%+ conversion rate free→premium"

- "80%+ D1 retention rate"

- "4.5+ rating nas lojas"

// Arquivo principal do aplicativo, onde a execução começa.

import 'package:flutter/material.dart';

// O ponto de entrada principal do seu aplicativo Flutter.

void main() {

// Inicia o aplicativo e carrega o widget principal MyApp.

runApp(const MyApp());

}

// O widget principal do aplicativo.

// Ele define o tema do app e a tela inicial.

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

// Retorna um MaterialApp para configurar a aplicação.

return MaterialApp(

title: 'Vibea+',

debugShowCheckedModeBanner: false, // Oculta a faixa de debug.

theme: ThemeData(

// Define o tema escuro como padrão, conforme o seu plano de design.

brightness: Brightness.dark,

scaffoldBackgroundColor: const Color(0xFF000000), // Preto base.

cardColor: const Color(0xFF0B0B0B),

// Define um esquema de cores personalizado para o tema escuro.

colorScheme: const ColorScheme.dark(

primary: Color(0xFF7C5CFC), // Cor de destaque roxo.

onPrimary: Color(0xFFFFFFFF),

secondary: Color(0xFFB3B3B3), // Cor secundária para textos.

surface: Color(0xFF121212), // Cor de superfície.

),

// Define o estilo do texto padrão para o tema.

textTheme: const TextTheme(

bodyLarge: TextStyle(color: Color(0xFFFFFFFF)),

bodyMedium: TextStyle(color: Color(0xFFB3B3B3)),

titleLarge: TextStyle(

color: Color(0xFFFFFFFF),

fontWeight: FontWeight.bold,

),

),

useMaterial3: true,

),

// A tela inicial do seu aplicativo será a de autenticação.

home: const AuthScreen(),

);

}

}

// A tela de Autenticação inicial do aplicativo.

// Esta é a primeira tela que o usuário verá.

class AuthScreen extends StatelessWidget {

const AuthScreen({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Padding(

padding: const EdgeInsets.symmetric(horizontal: 24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

// Título do aplicativo.

Text(

'Vibea+',

style: Theme.of(context).textTheme.titleLarge?.copyWith(fontSize: 48),

),

const SizedBox(height: 16.0),

// Descrição do app.

const Text(

'Bem-vindo! Onde seus caminhos se cruzam com as pessoas certas.',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 18.0),

),

const SizedBox(height: 48.0),

// Card com estilo de Neumorfismo para o login.

Container(

decoration: BoxDecoration(

color: Theme.of(context).cardColor,

borderRadius: BorderRadius.circular(12.0),

boxShadow: [

// Sombra superior/esquerda para efeito de relevo.

BoxShadow(

color: Colors.white.withOpacity(0.05),

offset: const Offset(-4, -4),

blurRadius: 8,

spreadRadius: 1,

),

// Sombra inferior/direita para efeito de extrusão.

BoxShadow(

color: Colors.black.withOpacity(0.4),

offset: const Offset(4, 4),

blurRadius: 8,

spreadRadius: 1,

),

],

),

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

children: [

Text(

'Autenticação',

style: Theme.of(context).textTheme.bodyLarge?.copyWith(

fontSize: 20,

fontWeight: FontWeight.bold,

),

),

const SizedBox(height: 24.0),

// Botão para o fluxo de login com telefone e SMS.

ElevatedButton.icon(

onPressed: () {

// TODO: Implementar a lógica de login por telefone.

// Você pode usar o pacote 'firebase\_auth' aqui.

},

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Theme.of(context).colorScheme.primary,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.phone),

label: const Text(

'Login por Telefone',

style: TextStyle(fontSize: 16),

),

),

],

),

),

),

],

),

),

),

);

}

}

// Arquivo principal do aplicativo, onde a execução começa.

import 'package:flutter/material.dart';

// O ponto de entrada principal do seu aplicativo Flutter.

void main() {

// Inicia o aplicativo e carrega o widget principal MyApp.

runApp(const MyApp());

}

// O widget principal do aplicativo.

// Ele define o tema do app e a tela inicial.

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

// Retorna um MaterialApp para configurar a aplicação.

return MaterialApp(

title: 'Vibea+',

debugShowCheckedModeBanner: false, // Oculta a faixa de debug.

theme: ThemeData(

// Define o tema escuro como padrão, conforme o seu plano de design.

brightness: Brightness.dark,

scaffoldBackgroundColor: const Color(0xFF000000), // Preto base.

cardColor: const Color(0xFF0B0B0B),

// Define um esquema de cores personalizado para o tema escuro.

colorScheme: const ColorScheme.dark(

primary: Color(0xFF7C5CFC), // Cor de destaque roxo.

onPrimary: Color(0xFFFFFFFF),

secondary: Color(0xFFB3B3B3), // Cor secundária para textos.

surface: Color(0xFF121212), // Cor de superfície.

),

// Define o estilo do texto padrão para o tema.

textTheme: const TextTheme(

bodyLarge: TextStyle(color: Color(0xFFFFFFFF)),

bodyMedium: TextStyle(color: Color(0xFFB3B3B3)),

titleLarge: TextStyle(

color: Color(0xFFFFFFFF),

fontWeight: FontWeight.bold,

),

),

useMaterial3: true,

),

// A tela inicial do seu aplicativo será a de autenticação.

home: const AuthScreen(),

);

}

}

// A tela de Autenticação inicial do aplicativo.

// Esta é a primeira tela que o usuário verá.

class AuthScreen extends StatelessWidget {

const AuthScreen({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Padding(

padding: const EdgeInsets.symmetric(horizontal: 24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

// Título do aplicativo.

Text(

'Vibea+',

style: Theme.of(context).textTheme.titleLarge?.copyWith(fontSize: 48),

),

const SizedBox(height: 16.0),

// Descrição do app.

const Text(

'Bem-vindo! Onde seus caminhos se cruzam com as pessoas certas.',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 18.0),

),

const SizedBox(height: 48.0),

// Card com estilo de Neumorfismo para o login.

Container(

decoration: BoxDecoration(

color: Theme.of(context).cardColor,

borderRadius: BorderRadius.circular(12.0),

boxShadow: [

// Sombra superior/esquerda para efeito de relevo.

BoxShadow(

color: Colors.white.withOpacity(0.05),

offset: const Offset(-4, -4),

blurRadius: 8,

spreadRadius: 1,

),

// Sombra inferior/direita para efeito de extrusão.

BoxShadow(

color: Colors.black.withOpacity(0.4),

offset: const Offset(4, 4),

blurRadius: 8,

spreadRadius: 1,

),

],

),

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

children: [

Text(

'Autenticação',

style: Theme.of(context).textTheme.bodyLarge?.copyWith(

fontSize: 20,

fontWeight: FontWeight.bold,

),

),

const SizedBox(height: 24.0),

// Botão para o fluxo de login com telefone e SMS.

ElevatedButton.icon(

onPressed: () {

// TODO: Implementar a lógica de login por telefone.

// Você pode usar o pacote 'firebase\_auth' aqui.

},

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Theme.of(context).colorScheme.primary,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.phone),

label: const Text(

'Login por Telefone',

style: TextStyle(fontSize: 16),

),

),

const SizedBox(height: 16.0),

// Botão para o fluxo de "Sign in with Google".

ElevatedButton.icon(

onPressed: () {

// TODO: Implementar a lógica de login com Google.

// Adicione a dependência `google\_sign\_in` ao seu projeto.

},

style: ElevatedButton.styleFrom(

foregroundColor: Colors.black,

backgroundColor: Colors.white,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.g\_mobiledata),

label: const Text(

'Entrar com Google',

style: TextStyle(fontSize: 16),

),

),

const SizedBox(height: 16.0),

// Botão para o fluxo de "Sign in with Apple".

ElevatedButton.icon(

onPressed: () {

// TODO: Implementar a lógica de login com Apple.

// Adicione a dependência `sign\_in\_with\_apple` ao seu projeto.

},

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Colors.black,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.apple),

label: const Text(

'Entrar com Apple',

style: TextStyle(fontSize: 16),

),

),

],

),

),

),

],

),

),

),

);

}

}

// Arquivo principal do aplicativo Vibea+.

// Este arquivo contém a inicialização do Firebase e a tela de autenticação.

import 'package:flutter/material.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:google\_sign\_in/google\_sign\_in.dart';

import 'package:sign\_in\_with\_apple/sign\_in\_with\_apple.dart';

// TODO: Certifique-se de adicionar as dependências no seu pubspec.yaml:

// firebase\_core: ^2.24.2

// firebase\_auth: ^4.15.1

// google\_sign\_in: ^6.1.6

// sign\_in\_with\_apple: ^4.3.0

// O ponto de entrada principal do seu aplicativo Flutter.

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(); // Inicializa o Firebase.

runApp(const MyApp());

}

// O widget principal do aplicativo.

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Vibea+',

debugShowCheckedModeBanner: false, // Oculta a faixa de debug.

theme: ThemeData(

brightness: Brightness.dark,

scaffoldBackgroundColor: const Color(0xFF000000), // Preto base.

cardColor: const Color(0xFF0B0B0B),

colorScheme: const ColorScheme.dark(

primary: Color(0xFF7C5CFC), // Roxo Vibrante

onPrimary: Color(0xFFFFFFFF),

secondary: Color(0xFFB3B3B3),

surface: Color(0xFF121212),

),

textTheme: const TextTheme(

bodyLarge: TextStyle(color: Color(0xFFFFFFFF)),

bodyMedium: TextStyle(color: Color(0xFFB3B3B3)),

titleLarge: TextStyle(

color: Color(0xFFFFFFFF),

fontWeight: FontWeight.bold,

),

),

useMaterial3: true,

),

home: AuthScreen(),

);

}

}

// A tela de Autenticação inicial do aplicativo.

class AuthScreen extends StatefulWidget {

const AuthScreen({super.key});

@override

State<AuthScreen> createState() => \_AuthScreenState();

}

class \_AuthScreenState extends State<AuthScreen> {

final FirebaseAuth \_auth = FirebaseAuth.instance;

final GoogleSignIn \_googleSignIn = GoogleSignIn();

// Função para exibir uma mensagem na parte inferior da tela.

void \_showMessage(String message) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text(message),

duration: const Duration(seconds: 2),

),

);

}

// Autenticação com Google

Future<void> \_signInWithGoogle() async {

try {

final GoogleSignInAccount? googleUser = await \_googleSignIn.signIn();

if (googleUser == null) {

\_showMessage('Login com Google cancelado.');

return;

}

final GoogleSignInAuthentication googleAuth = await googleUser.authentication;

final credential = GoogleAuthProvider.credential(

accessToken: googleAuth.accessToken,

idToken: googleAuth.idToken,

);

await \_auth.signInWithCredential(credential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com Google: ${e.message}');

} catch (e) {

\_showMessage('Erro inesperado: $e');

}

}

// Autenticação com Apple

Future<void> \_signInWithApple() async {

try {

final credential = await SignInWithApple.getAppleIDCredential(

scopes: [

AppleIDAuthorizationScopes.email,

AppleIDAuthorizationScopes.fullName,

],

);

final authCredential = AppleAuthProvider.credential(

identityToken: credential.identityToken,

rawNonce: credential.nonce,

);

await \_auth.signInWithCredential(authCredential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com Apple: ${e.message}');

} catch (e) {

\_showMessage('Erro inesperado: $e');

}

}

// Função para navegar para a próxima tela após o login.

void \_navigateToHome() {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const HomeScreen()),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Padding(

padding: const EdgeInsets.symmetric(horizontal: 24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Vibea+',

style: Theme.of(context).textTheme.titleLarge?.copyWith(fontSize: 48),

),

const SizedBox(height: 16.0),

const Text(

'Bem-vindo! Onde seus caminhos se cruzam com as pessoas certas.',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 18.0),

),

const SizedBox(height: 48.0),

Container(

decoration: BoxDecoration(

color: Theme.of(context).cardColor,

borderRadius: BorderRadius.circular(12.0),

boxShadow: [

BoxShadow(

color: Colors.white.withOpacity(0.05),

offset: const Offset(-4, -4),

blurRadius: 8,

spreadRadius: 1,

),

BoxShadow(

color: Colors.black.withOpacity(0.4),

offset: const Offset(4, 4),

blurRadius: 8,

spreadRadius: 1,

),

],

),

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

children: [

Text(

'Autenticação',

style: Theme.of(context).textTheme.bodyLarge?.copyWith(

fontSize: 20,

fontWeight: FontWeight.bold,

),

),

const SizedBox(height: 24.0),

ElevatedButton.icon(

onPressed: () {

// TODO: Implementar a lógica de login por telefone.

// A lógica do Firebase Phone Auth deve ser adicionada aqui.

\_showMessage('Funcionalidade de login por telefone em desenvolvimento.');

},

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Theme.of(context).colorScheme.primary,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.phone),

label: const Text(

'Login por Telefone',

style: TextStyle(fontSize: 16),

),

),

const SizedBox(height: 16.0),

ElevatedButton.icon(

onPressed: \_signInWithGoogle,

style: ElevatedButton.styleFrom(

foregroundColor: Colors.black,

backgroundColor: Colors.white,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.g\_mobiledata),

label: const Text(

'Entrar com Google',

style: TextStyle(fontSize: 16),

),

),

const SizedBox(height: 16.0),

if (Theme.of(context).platform == TargetPlatform.iOS)

ElevatedButton.icon(

onPressed: \_signInWithApple,

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Colors.black,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.apple),

label: const Text(

'Entrar com Apple',

style: TextStyle(fontSize: 16),

),

),

],

),

),

),

],

),

),

),

);

}

}

// Uma tela de placeholder para simular a navegação após o login.

class HomeScreen extends StatelessWidget {

const HomeScreen({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Vibea+'),

backgroundColor: Colors.transparent,

elevation: 0,

actions: [

IconButton(

icon: const Icon(Icons.logout),

onPressed: () async {

await FirebaseAuth.instance.signOut();

await GoogleSignIn().signOut();

if (context.mounted) {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const AuthScreen()),

);

}

},

),

],

),

body: const Center(

child: Text(

'Login realizado com sucesso!\nBem-vindo ao Vibea+.',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),

),

),

);

}

}

// Arquivo principal do aplicativo Vibea+.

// Este arquivo contém a inicialização do Firebase e a tela de autenticação.

import 'package:flutter/material.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:google\_sign\_in/google\_sign\_in.dart';

import 'package:sign\_in\_with\_apple/sign\_in\_with\_apple.dart';

// TODO: Certifique-se de adicionar as dependências no seu pubspec.yaml:

// firebase\_core: ^2.24.2

// firebase\_auth: ^4.15.1

// google\_sign\_in: ^6.1.6

// sign\_in\_with\_apple: ^4.3.0

// O ponto de entrada principal do seu aplicativo Flutter.

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(); // Inicializa o Firebase.

runApp(const MyApp());

}

// O widget principal do aplicativo.

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Vibea+',

debugShowCheckedModeBanner: false, // Oculta a faixa de debug.

theme: ThemeData(

brightness: Brightness.dark,

scaffoldBackgroundColor: const Color(0xFF000000), // Preto base.

cardColor: const Color(0xFF0B0B0B),

colorScheme: const ColorScheme.dark(

primary: Color(0xFF7C5CFC), // Roxo Vibrante

onPrimary: Color(0xFFFFFFFF),

secondary: Color(0xFFB3B3B3),

surface: Color(0xFF121212),

),

textTheme: const TextTheme(

bodyLarge: TextStyle(color: Color(0xFFFFFFFF)),

bodyMedium: TextStyle(color: Color(0xFFB3B3B3)),

titleLarge: TextStyle(

color: Color(0xFFFFFFFF),

fontWeight: FontWeight.bold,

),

),

useMaterial3: true,

),

home: AuthScreen(),

);

}

}

// A tela de Autenticação inicial do aplicativo.

class AuthScreen extends StatefulWidget {

const AuthScreen({super.key});

@override

State<AuthScreen> createState() => \_AuthScreenState();

}

class \_AuthScreenState extends State<AuthScreen> {

final FirebaseAuth \_auth = FirebaseAuth.instance;

final GoogleSignIn \_googleSignIn = GoogleSignIn();

// Função para exibir uma mensagem na parte inferior da tela.

void \_showMessage(String message) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text(message),

duration: const Duration(seconds: 2),

),

);

}

// Autenticação com Google

Future<void> \_signInWithGoogle() async {

try {

final GoogleSignInAccount? googleUser = await \_googleSignIn.signIn();

if (googleUser == null) {

\_showMessage('Login com Google cancelado.');

return;

}

final GoogleSignInAuthentication googleAuth = await googleUser.authentication;

final credential = GoogleAuthProvider.credential(

accessToken: googleAuth.accessToken,

idToken: googleAuth.idToken,

);

await \_auth.signInWithCredential(credential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com Google: ${e.message}');

} catch (e) {

\_showMessage('Erro inesperado: $e');

}

}

// Autenticação com Apple

Future<void> \_signInWithApple() async {

try {

final credential = await SignInWithApple.getAppleIDCredential(

scopes: [

AppleIDAuthorizationScopes.email,

AppleIDAuthorizationScopes.fullName,

],

);

final authCredential = AppleAuthProvider.credential(

identityToken: credential.identityToken,

rawNonce: credential.nonce,

);

await \_auth.signInWithCredential(authCredential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com Apple: ${e.message}');

} catch (e) {

\_showMessage('Erro inesperado: $e');

}

}

// Função para navegar para a próxima tela após o login.

void \_navigateToHome() {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const HomeScreen()),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Padding(

padding: const EdgeInsets.symmetric(horizontal: 24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Vibea+',

style: Theme.of(context).textTheme.titleLarge?.copyWith(fontSize: 48),

),

const SizedBox(height: 16.0),

const Text(

'Bem-vindo! Onde seus caminhos se cruzam com as pessoas certas.',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 18.0),

),

const SizedBox(height: 48.0),

Container(

decoration: BoxDecoration(

color: Theme.of(context).cardColor,

borderRadius: BorderRadius.circular(12.0),

boxShadow: [

BoxShadow(

color: Colors.white.withOpacity(0.05),

offset: const Offset(-4, -4),

blurRadius: 8,

spreadRadius: 1,

),

BoxShadow(

color: Colors.black.withOpacity(0.4),

offset: const Offset(4, 4),

blurRadius: 8,

spreadRadius: 1,

),

],

),

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

children: [

Text(

'Autenticação',

style: Theme.of(context).textTheme.bodyLarge?.copyWith(

fontSize: 20,

fontWeight: FontWeight.bold,

),

),

const SizedBox(height: 24.0),

ElevatedButton.icon(

onPressed: () {

// TODO: Implementar a lógica de login por telefone.

// A lógica do Firebase Phone Auth deve ser adicionada aqui.

\_navigateToPhoneAuth();

},

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Theme.of(context).colorScheme.primary,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.phone),

label: const Text(

'Login por Telefone',

style: TextStyle(fontSize: 16),

),

),

const SizedBox(height: 16.0),

ElevatedButton.icon(

onPressed: \_signInWithGoogle,

style: ElevatedButton.styleFrom(

foregroundColor: Colors.black,

backgroundColor: Colors.white,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.g\_mobiledata),

label: const Text(

'Entrar com Google',

style: TextStyle(fontSize: 16),

),

),

const SizedBox(height: 16.0),

if (Theme.of(context).platform == TargetPlatform.iOS)

ElevatedButton.icon(

onPressed: \_signInWithApple,

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Colors.black,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.apple),

label: const Text(

'Entrar com Apple',

style: TextStyle(fontSize: 16),

),

),

],

),

),

),

],

),

),

),

);

}

// Navega para a tela de autenticação por telefone

void \_navigateToPhoneAuth() {

Navigator.of(context).push(

MaterialPageRoute(builder: (\_) => const PhoneAuthScreen()),

);

}

}

// Uma tela de placeholder para simular a navegação após o login.

class HomeScreen extends StatelessWidget {

const HomeScreen({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Vibea+'),

backgroundColor: Colors.transparent,

elevation: 0,

actions: [

IconButton(

icon: const Icon(Icons.logout),

onPressed: () async {

await FirebaseAuth.instance.signOut();

await GoogleSignIn().signOut();

if (context.mounted) {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const AuthScreen()),

);

}

},

),

],

),

body: const Center(

child: Text(

'Login realizado com sucesso!\nBem-vindo ao Vibea+.',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),

),

),

);

}

}

// Nova tela para autenticação por telefone.

class PhoneAuthScreen extends StatefulWidget {

const PhoneAuthScreen({super.key});

@override

State<PhoneAuthScreen> createState() => \_PhoneAuthScreenState();

}

class \_PhoneAuthScreenState extends State<PhoneAuthScreen> {

final FirebaseAuth \_auth = FirebaseAuth.instance;

final TextEditingController \_phoneController = TextEditingController();

final TextEditingController \_otpController = TextEditingController();

String \_verificationId = '';

bool \_otpSent = false;

void \_showMessage(String message) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text(message),

duration: const Duration(seconds: 2),

),

);

}

Future<void> \_verifyPhoneNumber() async {

try {

await \_auth.verifyPhoneNumber(

phoneNumber: \_phoneController.text,

verificationCompleted: (PhoneAuthCredential credential) async {

await \_auth.signInWithCredential(credential);

\_navigateToHome();

},

verificationFailed: (FirebaseAuthException e) {

\_showMessage('Verificação falhou: ${e.message}');

},

codeSent: (String verificationId, int? resendToken) {

\_verificationId = verificationId;

setState(() {

\_otpSent = true;

});

\_showMessage('Código de verificação enviado.');

},

codeAutoRetrievalTimeout: (String verificationId) {

\_verificationId = verificationId;

},

timeout: const Duration(seconds: 60),

);

} catch (e) {

\_showMessage('Erro ao verificar o número: $e');

}

}

Future<void> \_signInWithOtp() async {

try {

final credential = PhoneAuthProvider.credential(

verificationId: \_verificationId,

smsCode: \_otpController.text,

);

await \_auth.signInWithCredential(credential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com o código: ${e.message}');

}

}

void \_navigateToHome() {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const HomeScreen()),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Login por Telefone'),

backgroundColor: Colors.transparent,

elevation: 0,

),

body: Center(

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

TextField(

controller: \_phoneController,

keyboardType: TextInputType.phone,

decoration: InputDecoration(

labelText: 'Número de Telefone',

hintText: '+55 11 99999-9999',

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

),

),

const SizedBox(height: 16.0),

ElevatedButton(

onPressed: \_verifyPhoneNumber,

style: ElevatedButton.styleFrom(

minimumSize: const Size(double.infinity, 50),

),

child: const Text('Enviar Código de Verificação'),

),

if (\_otpSent) ...[

const SizedBox(height: 24.0),

TextField(

controller: \_otpController,

keyboardType: TextInputType.number,

decoration: InputDecoration(

labelText: 'Código de Verificação (OTP)',

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

),

),

const SizedBox(height: 16.0),

ElevatedButton(

onPressed: \_signInWithOtp,

style: ElevatedButton.styleFrom(

minimumSize: const Size(double.infinity, 50),

backgroundColor: Theme.of(context).colorScheme.primary,

foregroundColor: Theme.of(context).colorScheme.onPrimary,

),

child: const Text('Entrar'),

),

],

],

),

),

),

);

}

}

// Arquivo principal do aplicativo Vibea+.

// Este arquivo contém a inicialização do Firebase e a tela de autenticação.

import 'package:flutter/material.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:google\_sign\_in/google\_sign\_in.dart';

import 'package:sign\_in\_with\_apple/sign\_in\_with\_apple.dart';

// TODO: Certifique-se de adicionar as dependências no seu pubspec.yaml:

// firebase\_core: ^2.24.2

// firebase\_auth: ^4.15.1

// google\_sign\_in: ^6.1.6

// sign\_in\_with\_apple: ^4.3.0

// O ponto de entrada principal do seu aplicativo Flutter.

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(); // Inicializa o Firebase.

runApp(const MyApp());

}

// O widget principal do aplicativo.

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Vibea+',

debugShowCheckedModeBanner: false, // Oculta a faixa de debug.

theme: ThemeData(

brightness: Brightness.dark,

scaffoldBackgroundColor: const Color(0xFF000000), // Preto base.

cardColor: const Color(0xFF0B0B0B),

colorScheme: const ColorScheme.dark(

primary: Color(0xFF7C5CFC), // Roxo Vibrante

onPrimary: Color(0xFFFFFFFF),

secondary: Color(0xFFB3B3B3),

surface: Color(0xFF121212),

),

textTheme: const TextTheme(

bodyLarge: TextStyle(color: Color(0xFFFFFFFF)),

bodyMedium: TextStyle(color: Color(0xFFB3B3B3)),

titleLarge: TextStyle(

color: Color(0xFFFFFFFF),

fontWeight: FontWeight.bold,

),

),

useMaterial3: true,

),

home: AuthScreen(),

);

}

}

// A tela de Autenticação inicial do aplicativo.

class AuthScreen extends StatefulWidget {

const AuthScreen({super.key});

@override

State<AuthScreen> createState() => \_AuthScreenState();

}

class \_AuthScreenState extends State<AuthScreen> {

final FirebaseAuth \_auth = FirebaseAuth.instance;

final GoogleSignIn \_googleSignIn = GoogleSignIn();

// Função para exibir uma mensagem na parte inferior da tela.

void \_showMessage(String message) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text(message),

duration: const Duration(seconds: 2),

),

);

}

// Autenticação com Google

Future<void> \_signInWithGoogle() async {

try {

final GoogleSignInAccount? googleUser = await \_googleSignIn.signIn();

if (googleUser == null) {

\_showMessage('Login com Google cancelado.');

return;

}

final GoogleSignInAuthentication googleAuth = await googleUser.authentication;

final credential = GoogleAuthProvider.credential(

accessToken: googleAuth.accessToken,

idToken: googleAuth.idToken,

);

await \_auth.signInWithCredential(credential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com Google: ${e.message}');

} catch (e) {

\_showMessage('Erro inesperado: $e');

}

}

// Autenticação com Apple

Future<void> \_signInWithApple() async {

try {

final credential = await SignInWithApple.getAppleIDCredential(

scopes: [

AppleIDAuthorizationScopes.email,

AppleIDAuthorizationScopes.fullName,

],

);

final authCredential = AppleAuthProvider.credential(

identityToken: credential.identityToken,

rawNonce: credential.nonce,

);

await \_auth.signInWithCredential(authCredential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com Apple: ${e.message}');

} catch (e) {

\_showMessage('Erro inesperado: $e');

}

}

// Função para navegar para a próxima tela após o login.

void \_navigateToHome() {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const HomeScreen()),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Padding(

padding: const EdgeInsets.symmetric(horizontal: 24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Vibea+',

style: Theme.of(context).textTheme.titleLarge?.copyWith(fontSize: 48),

),

const SizedBox(height: 16.0),

const Text(

'Bem-vindo! Onde seus caminhos se cruzam com as pessoas certas.',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 18.0),

),

const SizedBox(height: 48.0),

Container(

decoration: BoxDecoration(

color: Theme.of(context).cardColor,

borderRadius: BorderRadius.circular(12.0),

boxShadow: [

BoxShadow(

color: Colors.white.withOpacity(0.05),

offset: const Offset(-4, -4),

blurRadius: 8,

spreadRadius: 1,

),

BoxShadow(

color: Colors.black.withOpacity(0.4),

offset: const Offset(4, 4),

blurRadius: 8,

spreadRadius: 1,

),

],

),

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

children: [

Text(

'Autenticação',

style: Theme.of(context).textTheme.bodyLarge?.copyWith(

fontSize: 20,

fontWeight: FontWeight.bold,

),

),

const SizedBox(height: 24.0),

ElevatedButton.icon(

onPressed: () {

\_navigateToPhoneAuth();

},

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Theme.of(context).colorScheme.primary,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.phone),

label: const Text(

'Login por Telefone',

style: TextStyle(fontSize: 16),

),

),

const SizedBox(height: 16.0),

ElevatedButton.icon(

onPressed: \_signInWithGoogle,

style: ElevatedButton.styleFrom(

foregroundColor: Colors.black,

backgroundColor: Colors.white,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.g\_mobiledata),

label: const Text(

'Entrar com Google',

style: TextStyle(fontSize: 16),

),

),

const SizedBox(height: 16.0),

if (Theme.of(context).platform == TargetPlatform.iOS)

ElevatedButton.icon(

onPressed: \_signInWithApple,

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Colors.black,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.apple),

label: const Text(

'Entrar com Apple',

style: TextStyle(fontSize: 16),

),

),

],

),

),

),

],

),

),

),

);

}

// Navega para a tela de autenticação por telefone

void \_navigateToPhoneAuth() {

Navigator.of(context).push(

MaterialPageRoute(builder: (\_) => const PhoneAuthScreen()),

);

}

}

// Uma tela de placeholder para simular a navegação após o login.

class HomeScreen extends StatelessWidget {

const HomeScreen({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Vibea+'),

backgroundColor: Colors.transparent,

elevation: 0,

actions: [

IconButton(

icon: const Icon(Icons.account\_circle),

onPressed: () {

Navigator.of(context).push(

MaterialPageRoute(builder: (\_) => const ProfileScreen()),

);

},

),

IconButton(

icon: const Icon(Icons.logout),

onPressed: () async {

await FirebaseAuth.instance.signOut();

await GoogleSignIn().signOut();

if (context.mounted) {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const AuthScreen()),

);

}

},

),

],

),

body: const Center(

child: Text(

'Login realizado com sucesso!\nBem-vindo ao Vibea+.',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),

),

),

);

}

}

// Nova tela para autenticação por telefone.

class PhoneAuthScreen extends StatefulWidget {

const PhoneAuthScreen({super.key});

@override

State<PhoneAuthScreen> createState() => \_PhoneAuthScreenState();

}

class \_PhoneAuthScreenState extends State<PhoneAuthScreen> {

final FirebaseAuth \_auth = FirebaseAuth.instance;

final TextEditingController \_phoneController = TextEditingController();

final TextEditingController \_otpController = TextEditingController();

String \_verificationId = '';

bool \_otpSent = false;

void \_showMessage(String message) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text(message),

duration: const Duration(seconds: 2),

),

);

}

Future<void> \_verifyPhoneNumber() async {

try {

await \_auth.verifyPhoneNumber(

phoneNumber: \_phoneController.text,

verificationCompleted: (PhoneAuthCredential credential) async {

await \_auth.signInWithCredential(credential);

\_navigateToHome();

},

verificationFailed: (FirebaseAuthException e) {

\_showMessage('Verificação falhou: ${e.message}');

},

codeSent: (String verificationId, int? resendToken) {

\_verificationId = verificationId;

setState(() {

\_otpSent = true;

});

\_showMessage('Código de verificação enviado.');

},

codeAutoRetrievalTimeout: (String verificationId) {

\_verificationId = verificationId;

},

timeout: const Duration(seconds: 60),

);

} catch (e) {

\_showMessage('Erro ao verificar o número: $e');

}

}

Future<void> \_signInWithOtp() async {

try {

final credential = PhoneAuthProvider.credential(

verificationId: \_verificationId,

smsCode: \_otpController.text,

);

await \_auth.signInWithCredential(credential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com o código: ${e.message}');

}

}

void \_navigateToHome() {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const HomeScreen()),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Login por Telefone'),

backgroundColor: Colors.transparent,

elevation: 0,

),

body: Center(

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

TextField(

controller: \_phoneController,

keyboardType: TextInputType.phone,

decoration: InputDecoration(

labelText: 'Número de Telefone',

hintText: '+55 11 99999-9999',

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

),

),

const SizedBox(height: 16.0),

ElevatedButton(

onPressed: \_verifyPhoneNumber,

style: ElevatedButton.styleFrom(

minimumSize: const Size(double.infinity, 50),

),

child: const Text('Enviar Código de Verificação'),

),

if (\_otpSent) ...[

const SizedBox(height: 24.0),

TextField(

controller: \_otpController,

keyboardType: TextInputType.number,

decoration: InputDecoration(

labelText: 'Código de Verificação (OTP)',

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

),

),

const SizedBox(height: 16.0),

ElevatedButton(

onPressed: \_signInWithOtp,

style: ElevatedButton.styleFrom(

minimumSize: const Size(double.infinity, 50),

backgroundColor: Theme.of(context).colorScheme.primary,

foregroundColor: Theme.of(context).colorScheme.onPrimary,

),

child: const Text('Entrar'),

),

],

],

),

),

),

);

}

}

// Nova tela de perfil do usuário.

class ProfileScreen extends StatelessWidget {

const ProfileScreen({super.key});

@override

Widget build(BuildContext context) {

final user = FirebaseAuth.instance.currentUser;

return Scaffold(

appBar: AppBar(

title: const Text('Meu Perfil'),

backgroundColor: Colors.transparent,

elevation: 0,

),

body: Center(

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const CircleAvatar(

radius: 60,

// TODO: Adicionar lógica para exibir a foto do perfil do usuário

child: Icon(Icons.person, size: 80),

),

const SizedBox(height: 24),

Text(

user?.displayName ?? 'Nome do Usuário',

style: Theme.of(context).textTheme.titleLarge?.copyWith(fontSize: 28),

textAlign: TextAlign.center,

),

const SizedBox(height: 8),

Text(

user?.email ?? 'email@example.com',

style: Theme.of(context).textTheme.bodyMedium?.copyWith(fontSize: 16),

textAlign: TextAlign.center,

),

const SizedBox(height: 48),

ElevatedButton.icon(

onPressed: () {

// TODO: Implementar a navegação para a tela de edição de perfil

// Aqui o usuário poderá atualizar o nome, bio, etc.

},

icon: const Icon(Icons.edit),

label: const Text('Editar Perfil'),

style: ElevatedButton.styleFrom(

minimumSize: const Size(double.infinity, 50),

backgroundColor: Theme.of(context).colorScheme.primary,

foregroundColor: Theme.of(context).colorScheme.onPrimary,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

),

),

],

),

),

),

);

}

}

// Arquivo principal do aplicativo Vibea+.

// Este arquivo contém a inicialização do Firebase e a tela de autenticação.

import 'package:flutter/material.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:google\_sign\_in/google\_sign\_in.dart';

import 'package:sign\_in\_with\_apple/sign\_in\_with\_apple.dart';

// TODO: Certifique-se de adicionar as dependências no seu pubspec.yaml:

// firebase\_core: ^2.24.2

// firebase\_auth: ^4.15.1

// google\_sign\_in: ^6.1.6

// sign\_in\_with\_apple: ^4.3.0

// O ponto de entrada principal do seu aplicativo Flutter.

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(); // Inicializa o Firebase.

runApp(const MyApp());

}

// O widget principal do aplicativo.

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Vibea+',

debugShowCheckedModeBanner: false, // Oculta a faixa de debug.

theme: ThemeData(

brightness: Brightness.dark,

scaffoldBackgroundColor: const Color(0xFF000000), // Preto base.

cardColor: const Color(0xFF0B0B0B),

colorScheme: const ColorScheme.dark(

primary: Color(0xFF7C5CFC), // Roxo Vibrante

onPrimary: Color(0xFFFFFFFF),

secondary: Color(0xFFB3B3B3),

surface: Color(0xFF121212),

),

textTheme: const TextTheme(

bodyLarge: TextStyle(color: Color(0xFFFFFFFF)),

bodyMedium: TextStyle(color: Color(0xFFB3B3B3)),

titleLarge: TextStyle(

color: Color(0xFFFFFFFF),

fontWeight: FontWeight.bold,

),

),

useMaterial3: true,

),

home: AuthScreen(),

);

}

}

// A tela de Autenticação inicial do aplicativo.

class AuthScreen extends StatefulWidget {

const AuthScreen({super.key});

@override

State<AuthScreen> createState() => \_AuthScreenState();

}

class \_AuthScreenState extends State<AuthScreen> {

final FirebaseAuth \_auth = FirebaseAuth.instance;

final GoogleSignIn \_googleSignIn = GoogleSignIn();

// Função para exibir uma mensagem na parte inferior da tela.

void \_showMessage(String message) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text(message),

duration: const Duration(seconds: 2),

),

);

}

// Autenticação com Google

Future<void> \_signInWithGoogle() async {

try {

final GoogleSignInAccount? googleUser = await \_googleSignIn.signIn();

if (googleUser == null) {

\_showMessage('Login com Google cancelado.');

return;

}

final GoogleSignInAuthentication googleAuth = await googleUser.authentication;

final credential = GoogleAuthProvider.credential(

accessToken: googleAuth.accessToken,

idToken: googleAuth.idToken,

);

await \_auth.signInWithCredential(credential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com Google: ${e.message}');

} catch (e) {

\_showMessage('Erro inesperado: $e');

}

}

// Autenticação com Apple

Future<void> \_signInWithApple() async {

try {

final credential = await SignInWithApple.getAppleIDCredential(

scopes: [

AppleIDAuthorizationScopes.email,

AppleIDAuthorizationScopes.fullName,

],

);

final authCredential = AppleAuthProvider.credential(

identityToken: credential.identityToken,

rawNonce: credential.nonce,

);

await \_auth.signInWithCredential(authCredential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com Apple: ${e.message}');

} catch (e) {

\_showMessage('Erro inesperado: $e');

}

}

// Função para navegar para a próxima tela após o login.

void \_navigateToHome() {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const HomeScreen()),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Padding(

padding: const EdgeInsets.symmetric(horizontal: 24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Vibea+',

style: Theme.of(context).textTheme.titleLarge?.copyWith(fontSize: 48),

),

const SizedBox(height: 16.0),

const Text(

'Bem-vindo! Onde seus caminhos se cruzam com as pessoas certas.',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 18.0),

),

const SizedBox(height: 48.0),

Container(

decoration: BoxDecoration(

color: Theme.of(context).cardColor,

borderRadius: BorderRadius.circular(12.0),

boxShadow: [

BoxShadow(

color: Colors.white.withOpacity(0.05),

offset: const Offset(-4, -4),

blurRadius: 8,

spreadRadius: 1,

),

BoxShadow(

color: Colors.black.withOpacity(0.4),

offset: const Offset(4, 4),

blurRadius: 8,

spreadRadius: 1,

),

],

),

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

children: [

Text(

'Autenticação',

style: Theme.of(context).textTheme.bodyLarge?.copyWith(

fontSize: 20,

fontWeight: FontWeight.bold,

),

),

const SizedBox(height: 24.0),

ElevatedButton.icon(

onPressed: () {

\_navigateToPhoneAuth();

},

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Theme.of(context).colorScheme.primary,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.phone),

label: const Text(

'Login por Telefone',

style: TextStyle(fontSize: 16),

),

),

const SizedBox(height: 16.0),

ElevatedButton.icon(

onPressed: \_signInWithGoogle,

style: ElevatedButton.styleFrom(

foregroundColor: Colors.black,

backgroundColor: Colors.white,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.g\_mobiledata),

label: const Text(

'Entrar com Google',

style: TextStyle(fontSize: 16),

),

),

const SizedBox(height: 16.0),

if (Theme.of(context).platform == TargetPlatform.iOS)

ElevatedButton.icon(

onPressed: \_signInWithApple,

style: ElevatedButton.styleFrom(

foregroundColor: Colors.white,

backgroundColor: Colors.black,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50),

),

icon: const Icon(Icons.apple),

label: const Text(

'Entrar com Apple',

style: TextStyle(fontSize: 16),

),

),

],

),

),

),

],

),

),

),

);

}

// Navega para a tela de autenticação por telefone

void \_navigateToPhoneAuth() {

Navigator.of(context).push(

MaterialPageRoute(builder: (\_) => const PhoneAuthScreen()),

);

}

}

// Uma tela de placeholder para simular a navegação após o login.

class HomeScreen extends StatelessWidget {

const HomeScreen({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Vibea+'),

backgroundColor: Colors.transparent,

elevation: 0,

actions: [

IconButton(

icon: const Icon(Icons.account\_circle),

onPressed: () {

Navigator.of(context).push(

MaterialPageRoute(builder: (\_) => const ProfileScreen()),

);

},

),

IconButton(

icon: const Icon(Icons.logout),

onPressed: () async {

await FirebaseAuth.instance.signOut();

await GoogleSignIn().signOut();

if (context.mounted) {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const AuthScreen()),

);

}

},

),

],

),

body: const DiscoveryScreen(),

);

}

}

// Nova tela para autenticação por telefone.

class PhoneAuthScreen extends StatefulWidget {

const PhoneAuthScreen({super.key});

@override

State<PhoneAuthScreen> createState() => \_PhoneAuthScreenState();

}

class \_PhoneAuthScreenState extends State<PhoneAuthScreen> {

final FirebaseAuth \_auth = FirebaseAuth.instance;

final TextEditingController \_phoneController = TextEditingController();

final TextEditingController \_otpController = TextEditingController();

String \_verificationId = '';

bool \_otpSent = false;

void \_showMessage(String message) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text(message),

duration: const Duration(seconds: 2),

),

);

}

Future<void> \_verifyPhoneNumber() async {

try {

await \_auth.verifyPhoneNumber(

phoneNumber: \_phoneController.text,

verificationCompleted: (PhoneAuthCredential credential) async {

await \_auth.signInWithCredential(credential);

\_navigateToHome();

},

verificationFailed: (FirebaseAuthException e) {

\_showMessage('Verificação falhou: ${e.message}');

},

codeSent: (String verificationId, int? resendToken) {

\_verificationId = verificationId;

setState(() {

\_otpSent = true;

});

\_showMessage('Código de verificação enviado.');

},

codeAutoRetrievalTimeout: (String verificationId) {

\_verificationId = verificationId;

},

timeout: const Duration(seconds: 60),

);

} catch (e) {

\_showMessage('Erro ao verificar o número: $e');

}

}

Future<void> \_signInWithOtp() async {

try {

final credential = PhoneAuthProvider.credential(

verificationId: \_verificationId,

smsCode: \_otpController.text,

);

await \_auth.signInWithCredential(credential);

\_navigateToHome();

} on FirebaseAuthException catch (e) {

\_showMessage('Erro ao entrar com o código: ${e.message}');

}

}

void \_navigateToHome() {

Navigator.of(context).pushReplacement(

MaterialPageRoute(builder: (\_) => const HomeScreen()),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Login por Telefone'),

backgroundColor: Colors.transparent,

elevation: 0,

),

body: Center(

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

TextField(

controller: \_phoneController,

keyboardType: TextInputType.phone,

decoration: InputDecoration(

labelText: 'Número de Telefone',

hintText: '+55 11 99999-9999',

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

),

),

const SizedBox(height: 16.0),

ElevatedButton(

onPressed: \_verifyPhoneNumber,

style: ElevatedButton.styleFrom(

minimumSize: const Size(double.infinity, 50),

),

child: const Text('Enviar Código de Verificação'),

),

if (\_otpSent) ...[

const SizedBox(height: 24.0),

TextField(

controller: \_otpController,

keyboardType: TextInputType.number,

decoration: InputDecoration(

labelText: 'Código de Verificação (OTP)',

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

),

),

const SizedBox(height: 16.0),

ElevatedButton(

onPressed: \_signInWithOtp,

style: ElevatedButton.styleFrom(

minimumSize: const Size(double.infinity, 50),

backgroundColor: Theme.of(context).colorScheme.primary,

foregroundColor: Theme.of(context).colorScheme.onPrimary,

),

child: const Text('Entrar'),

),

],

],

),

),

),

);

}

}

// Nova tela de perfil do usuário.

class ProfileScreen extends StatelessWidget {

const ProfileScreen({super.key});

@override

Widget build(BuildContext context) {

final user = FirebaseAuth.instance.currentUser;

return Scaffold(

appBar: AppBar(

title: const Text('Meu Perfil'),

backgroundColor: Colors.transparent,

elevation: 0,

),

body: Center(

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const CircleAvatar(

radius: 60,

// TODO: Adicionar lógica para exibir a foto do perfil do usuário

child: Icon(Icons.person, size: 80),

),

const SizedBox(height: 24),

Text(

user?.displayName ?? 'Nome do Usuário',

style: Theme.of(context).textTheme.titleLarge?.copyWith(fontSize: 28),

textAlign: TextAlign.center,

),

const SizedBox(height: 8),

Text(

user?.email ?? 'email@example.com',

style: Theme.of(context).textTheme.bodyMedium?.copyWith(fontSize: 16),

textAlign: TextAlign.center,

),

const SizedBox(height: 48),

ElevatedButton.icon(

onPressed: () {

// TODO: Implementar a navegação para a tela de edição de perfil

// Aqui o usuário poderá atualizar o nome, bio, etc.

},

icon: const Icon(Icons.edit),

label: const Text('Editar Perfil'),

style: ElevatedButton.styleFrom(

minimumSize: const Size(double.infinity, 50),

backgroundColor: Theme.of(context).colorScheme.primary,

foregroundColor: Theme.of(context).colorScheme.onPrimary,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

),

),

],

),

),

),

);

}

}

// Classe para representar os dados de um usuário para a descoberta.

class UserProfile {

final String name;

final int age;

final String bio;

final String imageUrl;

UserProfile({

required this.name,

required this.age,

required this.bio,

required this.imageUrl,

});

}

// Nova tela para a funcionalidade de descoberta (swipe).

class DiscoveryScreen extends StatefulWidget {

const DiscoveryScreen({super.key});

@override

State<DiscoveryScreen> createState() => \_DiscoveryScreenState();

}

class \_DiscoveryScreenState extends State<DiscoveryScreen> {

final List<UserProfile> \_users = [

UserProfile(

name: 'Maria',

age: 25,

bio: 'Amante de café e longas caminhadas. Procurando por uma boa conversa.',

imageUrl: 'https://placehold.co/400x500/0B0B0B/B3B3B3?text=Maria',

),

UserProfile(

name: 'João',

age: 28,

bio: 'Desenvolvedor e entusiasta de tecnologia. Gosto de jogos e filmes de ficção científica.',

imageUrl: 'https://placehold.co/400x500/0B0B0B/B3B3B3?text=Joao',

),

UserProfile(

name: 'Ana',

age: 22,

bio: 'Artista e sonhadora. Adoro pintar e viajar.',

imageUrl: 'https://placehold.co/400x500/0B0B0B/B3B3B3?text=Ana',

),

];

int \_currentIndex = 0;

void \_onSwipeLeft() {

setState(() {

if (\_currentIndex < \_users.length - 1) {

\_currentIndex++;

} else {

\_showMessage('Fim da lista!');

// TODO: Implementar lógica para recarregar mais usuários

}

});

}

void \_onSwipeRight() {

setState(() {

if (\_currentIndex < \_users.length - 1) {

\_currentIndex++;

} else {

\_showMessage('Fim da lista!');

// TODO: Implementar lógica para recarregar mais usuários

}

});

// TODO: Adicionar lógica para "curtir" o usuário no backend (Firestore)

\_showMessage('Você curtiu o perfil de ${\_users[\_currentIndex - 1].name}');

}

void \_showMessage(String message) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text(message),

duration: const Duration(seconds: 2),

),

);

}

@override

Widget build(BuildContext context) {

return Center(

child: Padding(

padding: const EdgeInsets.symmetric(horizontal: 24.0, vertical: 32.0),

child: Column(

children: [

Expanded(

child: \_currentIndex < \_users.length

? UserCard(user: \_users[\_currentIndex])

: const Center(

child: Text(

'Não há mais perfis por perto.',

style: TextStyle(fontSize: 20),

textAlign: TextAlign.center,

),

),

),

const SizedBox(height: 24),

Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: [

FloatingActionButton(

onPressed: \_onSwipeLeft,

backgroundColor: const Color(0xFF1E1E1E),

child: const Icon(Icons.close, color: Colors.red),

),

FloatingActionButton(

onPressed: \_onSwipeRight,

backgroundColor: const Color(0xFF1E1E1E),

child: const Icon(Icons.favorite, color: Colors.green),

),

],

),

],

),

),

);

}

}

// Widget do cartão de usuário para o modo de descoberta.

class UserCard extends StatelessWidget {

final UserProfile user;

const UserCard({super.key, required this.user});

@override

Widget build(BuildContext context) {

return Card(

clipBehavior: Clip.antiAlias,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(16.0),

),

child: Stack(

fit: StackFit.expand,

children: [

Image.network(

user.imageUrl,

fit: BoxFit.cover,

),

Container(

decoration: BoxDecoration(

gradient: LinearGradient(

begin: Alignment.topCenter,

end: Alignment.bottomCenter,

colors: [

Colors.transparent,

Colors.black.withOpacity(0.7),

],

),

),

),

Positioned(

bottom: 16.0,

left: 16.0,

right: 16.0,

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

'${user.name}, ${user.age}',

style: const TextStyle(

color: Colors.white,

fontSize: 28,

fontWeight: FontWeight.bold,

),

),

const SizedBox(height: 8),

Text(

user.bio,

style: const TextStyle(

color: Colors.white,

fontSize: 16,

),

),

],

),

),

],

),

);

}

}