MainActivity.kt

package com.example.women\_safety\_app  
  
import io.flutter.embedding.android.FlutterActivity  
import io.flutter.embedding.engine.FlutterEngine  
import io.flutter.plugin.common.MethodChannel  
import android.content.Intent  
import android.net.Uri  
  
class MainActivity: FlutterActivity() {  
 private val CHANNEL = "com.example.women\_safety\_app/sms"  
  
 override fun configureFlutterEngine(flutterEngine: FlutterEngine) {  
 super.configureFlutterEngine(flutterEngine)  
  
 MethodChannel(flutterEngine.dartExecutor.binaryMessenger, CHANNEL).setMethodCallHandler {  
 call, result ->  
 if (call.method == "sendSMS") {  
 val phone = call.argument<String>("phone")  
 val message = call.argument<String>("message")  
 sendSMS(phone, message)  
 result.success(null)  
 } else {  
 result.notImplemented()  
 }  
 }  
 }  
  
 private fun sendSMS(phone: String?, message: String?) {  
 val uri = Uri.parse("smsto:$phone")  
 val intent = Intent(Intent.ACTION\_SENDTO, uri)  
 intent.putExtra("sms\_body", message)  
 intent.flags = Intent.FLAG\_ACTIVITY\_NEW\_TASK  
 startActivity(intent)  
 }  
}

home\_screen.dart

import 'package:flutter/material.dart';  
import 'package:geolocator/geolocator.dart';  
import 'package:permission\_handler/permission\_handler.dart';  
import 'package:flutter\_background\_messenger/flutter\_background\_messenger.dart';  
  
class HomeScreen extends StatelessWidget {  
 final FlutterBackgroundMessenger \_messenger = FlutterBackgroundMessenger();  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 backgroundColor: Colors.*white*,  
 appBar: AppBar(  
 title: Text("Women's Safety App", style: TextStyle(fontWeight: FontWeight.*bold*)),  
 backgroundColor: Colors.*pinkAccent*,  
 ),  
 body: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Padding(  
 padding: EdgeInsets.symmetric(horizontal: 20),  
 child: GridView.count(  
 shrinkWrap: true,  
 crossAxisCount: 2,  
 crossAxisSpacing: 20,  
 mainAxisSpacing: 20,  
 children: [  
 \_buildNavButton(context, "Emergency Contacts", Icons.*contact\_phone*, Colors.*blue*),  
 \_buildNavButton(context, "Community", Icons.*people*, Colors.*green*),  
 \_buildNavButton(context, "Timer", Icons.*timer*, Colors.*orange*),  
 \_buildNavButton(context, "Safety Tips", Icons.*security*, Colors.*purple*),  
 \_buildNavButton(context, "Priority Messaging", Icons.*priority\_high*, Colors.*red*),  
 ],  
 ),  
 ),  
 SizedBox(height: 30),  
 FloatingActionButton.extended(  
 onPressed: () async {  
 print("🆘 SOS Button Clicked!");  
 await sendSOS();  
 },  
 label: Text(  
 "SOS ALERT",  
 style: TextStyle(fontSize: 20, fontWeight: FontWeight.*bold*, color: Colors.*white*),  
 ),  
 icon: Icon(Icons.*sos*, size: 30, color: Colors.*white*),  
 backgroundColor: Colors.*red*,  
 elevation: 10,  
 ),  
 SizedBox(height: 20),  
 ],  
 ),  
 );  
 }  
  
 Widget \_buildNavButton(BuildContext context, String title, IconData icon, Color color) {  
 return ElevatedButton(  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: color,  
 shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(15)),  
 padding: EdgeInsets.all(15),  
 ),  
 onPressed: () {},  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Icon(icon, size: 40, color: Colors.*white*),  
 SizedBox(height: 10),  
 Text(  
 title,  
 textAlign: TextAlign.center,  
 style: TextStyle(color: Colors.*white*, fontSize: 16, fontWeight: FontWeight.*bold*),  
 ),  
 ],  
 ),  
 );  
 }  
  
 Future<void> sendSOS() async {  
 print("⚡ Attempting to send SMS...");  
  
 // Request necessary permissions  
 var locationStatus = await Permission.*location*.request();  
 var smsStatus = await Permission.*sms*.request();  
  
 if (!locationStatus.isGranted || !smsStatus.isGranted) {  
 print("❌ Permissions not granted. Cannot send SOS.");  
 return;  
 }  
  
 try {  
 // Get user location  
 print("📍 Fetching user location...");  
 Position position = await Geolocator.*getCurrentPosition*(desiredAccuracy: LocationAccuracy.high);  
 String googleMapsUrl =  
 "https://www.google.com/maps/search/?api=1&query=${position.latitude},${position.longitude}";  
 print("📍 Location Fetched: $googleMapsUrl");  
  
 // Prepare SOS message  
 String message = "🚨 SOS Alert! I'm in danger. My location: $googleMapsUrl";  
 String recipient = "9022302137"; // Emergency contact number  
  
 print("📞 Sending to: $recipient");  
 print("📩 Message content: $message");  
  
 // Send SMS using flutter\_background\_messenger  
 bool success = await \_messenger.sendSMS(  
 phoneNumber: recipient,  
 message: message,  
 );  
  
 if (success) {  
 print("✅ SMS sent successfully to $recipient");  
 } else {  
 print("❌ Failed to send SMS to $recipient");  
 }  
 } catch (e) {  
 print("❌ Error sending SOS: $e");  
 }  
 }  
}

pubspec.yaml

name: women\_safety\_app  
description: "A new Flutter project."  
*# The following line prevents the package from being accidentally published to  
# pub.dev using `flutter pub publish`. This is preferred for private packages.*publish\_to: 'none' *# Remove this line if you wish to publish to pub.dev  
  
# The following defines the version and build number for your application.  
# A version number is three numbers separated by dots, like 1.0.0  
# followed by an optional build number separated by a +.*version: 1.0.0+1  
  
*# Environment settings*environment:  
 sdk: '>=3.6.1 <4.0.0'  
  
*# Dependencies specify other packages that your package needs in order to work.*dependencies:  
 flutter:  
 sdk: flutter  
 geolocator: ^13.0.4 *# Latest resolvable* permission\_handler: ^11.4.0 *# Latest resolvable* flutter\_sms: ^2.3.3  
 *#telephony: ^0.2.0  
 # Cupertino Icons for iOS-style icons* flutter\_background\_messenger: ^0.0.2  
 cupertino\_icons: ^1.0.8  
  
*# Dev dependencies for testing and linting*dev\_dependencies:  
 flutter\_test:  
 sdk: flutter  
 flutter\_lints: ^5.0.0  
  
*# Flutter-specific configuration*flutter:  
 *# Ensures that Material Icons are available* uses-material-design: true  
  
 *# Add assets for images, icons, etc.* assets:  
 - lib/resources/assets/