RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

Internship Report

on

Mobile Application Development
INT410: Intra Institutional Internship

STUDENT NAME: SRISHTI
USN : 1MS24CS188

STUDENT NAME : TOOLIKA TRIPATHI
USN : 1MS24CS204

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)
MSR Nagar, MSRIT Post, Bangalore-560054

August - 2025

RAMAIAH
Institute of Technology

RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054

(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that Mr./Ms. Srishti (USN:1MS24CS188) and Mr./Ms. Toolika Tripathi(USN:1MS24CS204), students of Bachelor of Engineering, have successfully completed, 24 Hours: from 05.08.2025 to 12.08.2025 Intra Institutional Internship in Mobile Application Development from the Department of Computer Science & Engineering, M S Ramaiah Institute of Technology, Bangalore.

SL No.	Component	Maximum Marks	Marks Obtained
1	Continuous Evaluation	50	
2	Presentation	20	
3	Report	30	
Total Marks		100	

Signature of the Student with Date **Signature of the Faculty Co-Ordinator**

Signature of Head of the Department

OVERVIEW OF INTERNSHIP ACTIVITIES

DATE	DAY	NAME OF THE TOPIC COMPLETED
05.08.25	Tuesday	Basic of Dart + Coding in Dartpad + Basics of Flutter
06.08.25	Wednesday	Installation of Flutter + Extensions + 5 Exercise Questions
07.08.25	Thursday	Flutter Widgets + Types of Widgets + Flutter Scaffold + Flutter Container + Flutter Row &Column + Flutter Text + Flutter Buttons + Flutter alert dialogue + Flutter icons + Flutter images + Flutter lists + Flutter toast + Flutter themes + Flutter charts
09.08.25	Saturday	Flutter Lists + Flutter Animations + Resume in Flutter
11.08.25	Monday	Assignment + Firebase Setup
12.08.25	Tuesday	GitHub repository + Project Submission

TABLE OF CONTENTS

Overall View of the Elder Aid App Implementation

1. Introduction

The Elder Aid app was created to assist elderly users in requesting help easily and efficiently. Many elders face challenges when trying to communicate their needs or find volunteers nearby who can assist them promptly. Our goal was to build a simple, user-friendly app that allows elders to submit requests for assistance and view the status of their requests seamlessly. The app focuses on providing a smooth experience tailored specifically to elders' needs.

2. Understanding User Needs and Roles

We identified two primary user roles in the app:

- Elders: The main users who submit help requests, describe their needs, and track responses.
- Volunteers: (Not implemented in this version) would help fulfill requests in a future expansion.

Currently, the app focuses on the elder's side, simplifying their interaction to submitting requests and viewing statuses without additional complexity.

3. Key Features and User Interface Design

The app's main features include:

- Account creation and sign-in to secure user data.
- Ability to submit different types of help requests (normal and emergency).
- Automatic location attachment to requests for better volunteer assistance.
- Viewing a personalized list of all submitted requests with detailed statuses.
- A profile section showing user information and logout functionality.

We aimed for a clean, minimalistic UI with large buttons, clear text, and straightforward navigation, making the app accessible to users who may not be tech-savvy.

4. Technology Stack and Tools Used

- Flutter: For building a cross-platform mobile app working on both Android and iOS.
- Firebase Authentication: To manage user accounts securely.
- Cloud Firestore: As the backend database for storing user profiles and help requests.
- Geolocator plugin: To obtain and attach user location data to requests.

These technologies provided fast development cycles and real-time syncing of data, critical for responsive app behavior.

5. Development Process and Challenges

Initially, the app supported basic account creation and request submission with no location support. Over time, we added:

- Location permission handling and real-time location fetching.
- Categorization of requests by type (normal vs emergency) with visual indicators.
- Request lists with filtering to show only the current user's requests.
- Improvements in UI responsiveness and error handling.

Challenges included:

- Handling location permissions smoothly on different devices.
- Ensuring fast data loading and avoiding app freezes during network calls.
- Designing an intuitive experience for elderly users, minimizing input steps.

6. Iterative Improvements

Through testing and user feedback, we:

- Improved button sizes and text readability.
- Added loading indicators during network calls for better feedback.
- Optimized Firestore queries to fetch only relevant user data.
- Refined navigation so switching between requests and profile is seamless.

7. Final Outcome

The Elder Aid app now:

- Allows elders to create accounts and securely sign in.
- Lets users submit detailed help requests, attaching location automatically.
- Displays a clear list of all user requests with their current statuses.
- Provides a profile screen for user info and sign-out option.
- Works smoothly on both Android and iOS platforms.

This implementation supports elders in asking for help with minimal barriers, improving their access to timely assistance.

8. Future Enhancements

Planned improvements include:

- Adding a volunteer module to accept and manage requests.
- Integrating real-time request tracking and volunteer arrival estimates.
- Introducing chat or voice communication between elders and volunteers.
- Incorporating emergency alert features with SOS buttons.
- Enhancing accessibility with voice commands and screen reader support.

Code Of Main Modules:

```dart
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:url_launcher/url_launcher.dart';
import 'package:geolocator/geolocator.dart';

void main() {
  runApp(const ElderAidApp());
}

class ElderAidApp extends StatelessWidget {
  const ElderAidApp({super.key});

  static const String homeRoute = '/home';
  static const String sosRoute = '/sos';
  static const String tasksRoute = '/tasks';
  static const String contactRoute = '/contact';
  static const String aboutRoute = '/about';

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'ElderAid',
      theme: ThemeData(colorSchemeSeed: Colors.teal, useMaterial3: true),
      initialRoute: homeRoute,
      routes: {
        homeRoute: (context) => const HomeScreen(),
        sosRoute: (context) => const SosScreen(),
        tasksRoute: (context) => const TasksScreen(),
        contactRoute: (context) => const ContactAdminScreen(),
        aboutRoute: (context) => const AboutScreen(),
      },
    );
  }
}

class AppDrawer extends StatelessWidget {
  const AppDrawer({super.key});

  @override
  Widget build(BuildContext context) {
    return Drawer(
      child: ListView(
```

```
        children: [
          const DrawerHeader(
            decoration: BoxDecoration(color: Colors.teal),
            child: Text(
              'ElderAid',
              style: TextStyle(fontSize: 24, color: Colors.white),
            ),
          ),
          ListTile(
            leading: const Icon(Icons.home),
            title: const Text('Home'),
            onTap: () {
              Navigator.pushNamedAndRemoveUntil(
                context,
                ElderAidApp.homeRoute,
                (route) => false,
              );
            },
          ),
          ListTile(
            leading: const Icon(Icons.report_problem),
            title: const Text('SOS / Request Help'),
            onTap: () {
              Navigator.pushNamedAndRemoveUntil(
                context,
                ElderAidApp.sosRoute,
                (route) => false,
              );
            },
          ),
          ListTile(
            leading: const Icon(Icons.check_box),
            title: const Text('Reminders'),
            onTap: () {
              Navigator.pushNamedAndRemoveUntil(
                context,
                ElderAidApp.tasksRoute,
                (route) => false,
              );
            },
          ),
          ListTile(
            leading: const Icon(Icons.contact_mail),
            title: const Text('Contact Admin / Caregiver'),
            onTap: () {
```

```
              Navigator.pushNamedAndRemoveUntil(
                context,
                ElderAidApp.contactRoute,
                (route) => false,
              );
            },
          ),
          ListTile(
            leading: const Icon(Icons.info),
            title: const Text('About'),
            onTap: () {
              Navigator.pushNamedAndRemoveUntil(
                context,
                ElderAidApp.aboutRoute,
                (route) => false,
              );
            },
          ),
        ],
      ),
    );
  }
}

// Home Screen
class HomeScreen extends StatelessWidget {
  const HomeScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      drawer: const AppDrawer(),
      appBar: AppBar(title: const Text('ElderAid')),
      body: Container(
        color: Colors.teal.shade50,
        child: ListView(
          padding: const EdgeInsets.all(16),
          children: [
            ClipRRect(
              borderRadius: BorderRadius.circular(12),
              child: Image.network(
                'https://images.unsplash.com/photo-1507003211169-
0a1dd7228f2d?auto=format&fit=crop&w=800&q=80',
                height: 180,
                fit: BoxFit.cover,
```

```
              ),
            ),
            const SizedBox(height: 16),
            const Text(
              'Welcome to ElderAid',
              style: TextStyle(
                fontSize: 24,
                fontWeight: FontWeight.bold,
                color: Colors.teal,
              ),
              textAlign: TextAlign.center,
            ),
            const SizedBox(height: 12),
            const Text(
              'ElderAid provides a simple, reliable way to get assistance,
manage personal reminders, and contact caregivers or admin quickly.',
              style: TextStyle(fontSize: 16),
              textAlign: TextAlign.center,
            ),
            const SizedBox(height: 24),
            Card(
              elevation: 4,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10),
              ),
              child: ListTile(
                leading: const Icon(Icons.report_problem, color:
Colors.teal),
                title: const Text('SOS / Request Help'),
                trailing: const Icon(Icons.arrow_forward_ios),
                onTap: () => Navigator.pushNamed(context,
ElderAidApp.sosRoute),
              ),
            ),
            const SizedBox(height: 12),
            Card(
              elevation: 4,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10),
              ),
              child: ListTile(
                leading: const Icon(Icons.check_box, color: Colors.teal),
                title: const Text('Manage Reminders'),
                trailing: const Icon(Icons.arrow_forward_ios),
                onTap: () =>
```

```
                    Navigator.pushNamed(context, ElderAidApp.tasksRoute),
              ),
            ),
            const SizedBox(height: 12),
            Card(
              elevation: 4,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10),
              ),
              child: ListTile(
                leading: const Icon(Icons.contact_mail, color:
Colors.teal),
                title: const Text('Contact Caregiver / Admin'),
                trailing: const Icon(Icons.arrow_forward_ios),
                onTap: () =>
                    Navigator.pushNamed(context,
ElderAidApp.contactRoute),
              ),
            ),
          ],
        ),
      ),
    );
  }
}

// SOS Screen: Call, SMS, Email, Share Location
class SosScreen extends StatefulWidget {
  const SosScreen({super.key});

  @override
  State<SosScreen> createState() => _SosScreenState();
}

class _SosScreenState extends State<SosScreen> {
  bool gettingLocation = false;
  String? lastLocation;

  Future<void> _callEmergency() async {
    final uri = Uri.parse('tel:108'); // change to desired emergency
number
    if (await canLaunchUrl(uri)) {
      await launchUrl(uri);
    } else {
      _showSnack('Could not open dialer');
```

```
    }
  }

  Future<void> _sendSms() async {
    final body = Uri.encodeComponent('I need help. Please reach me.');
    final uri = Uri.parse('sms:?body=$body');
    if (await canLaunchUrl(uri)) {
      await launchUrl(uri);
    } else {
      _showSnack('Could not open SMS app');
    }
  }

  Future<void> _sendEmail({String? locationLink}) async {
    final subject = Uri.encodeComponent('Emergency Assistance Needed');
    final body = Uri.encodeComponent(
      'I need help.\n\nLocation: ${locationLink ?? "Not
provided"}\n\nPlease respond quickly.',
    );
    final mailto = Uri.parse(
      'mailto:caregiver@family.example?subject=$subject&body=$body',
    );
    if (await canLaunchUrl(mailto)) {
      await launchUrl(mailto);
    } else {
      _showSnack('Could not open mail app');
    }
  }

  Future<void> _shareLocation() async {
    setState(() {
      gettingLocation = true;
    });

    bool serviceEnabled;
    LocationPermission permission;

    serviceEnabled = await Geolocator.isLocationServiceEnabled();
    if (!serviceEnabled) {
      _showSnack('Location services are disabled. Please enable them.');
      setState(() => gettingLocation = false);
      return;
    }

    permission = await Geolocator.checkPermission();
```

```
    if (permission == LocationPermission.denied) {
      permission = await Geolocator.requestPermission();
      if (permission == LocationPermission.denied) {
        _showSnack('Location permission denied');
        setState(() => gettingLocation = false);
        return;
      }
    }

    if (permission == LocationPermission.deniedForever) {
      _showSnack(
        'Location permissions are permanently denied. Open app settings.',
      );
      setState(() => gettingLocation = false);
      return;
    }

    try {
      final position = await Geolocator.getCurrentPosition(
        desiredAccuracy: LocationAccuracy.high,
      );
      final link =
          'https://www.google.com/maps/search/?api=1&query=${position.lati
tude},${position.longitude}';
      setState(() {
        lastLocation = link;
        gettingLocation = false;
      });
      // After obtaining location, open email compose with link as quick
action
      _sendEmail(locationLink: link);
    } catch (e) {
      _showSnack('Could not obtain location');
      setState(() => gettingLocation = false);
    }
  }

  void _showSnack(String msg) {
    ScaffoldMessenger.of(context).showSnackBar(SnackBar(content:
Text(msg)));
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

```
      drawer: const AppDrawer(),
      appBar: AppBar(title: const Text('SOS / Request Help')),
      body: Padding(
        padding: const EdgeInsets.all(24),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            const Icon(Icons.report_problem, size: 88, color:
Colors.redAccent),
            const SizedBox(height: 16),
            const Text(
              'If you need immediate help, use one of the options below.',
              textAlign: TextAlign.center,
              style: TextStyle(fontSize: 18),
            ),
            const SizedBox(height: 24),
            ElevatedButton.icon(
              style: ElevatedButton.styleFrom(
                minimumSize: const Size.fromHeight(56),
                backgroundColor: Colors.redAccent,
              ),
              icon: const Icon(Icons.call),
              label: const Text('Call Emergency'),
              onPressed: _callEmergency,
            ),
            const SizedBox(height: 12),
            ElevatedButton.icon(
              style: ElevatedButton.styleFrom(
                minimumSize: const Size.fromHeight(56),
              ),
              icon: const Icon(Icons.sms),
              label: const Text('Send SMS Alert'),
              onPressed: _sendSms,
            ),
            const SizedBox(height: 12),
            ElevatedButton.icon(
              style: ElevatedButton.styleFrom(
                minimumSize: const Size.fromHeight(56),
              ),
              icon: const Icon(Icons.location_on),
              label: gettingLocation
                  ? const Text('Getting location...')
                  : const Text('Share Location & Email Caregiver'),
              onPressed: gettingLocation ? null : _shareLocation,
            ),
```

```
            if (lastLocation != null) ...[
              const SizedBox(height: 16),
              Text('Last shared location:'),
              SelectableText(lastLocation!),
            ],
          ],
        ),
      ),
    );
  }
}

// Tasks Screen with CRUD & SharedPreferences storage (Renamed Reminders)
class TasksScreen extends StatefulWidget {
  const TasksScreen({super.key});

  @override
  State<TasksScreen> createState() => _TasksScreenState();
}

class _TasksScreenState extends State<TasksScreen> {
  List<String> tasks = [];
  final _taskController = TextEditingController();
  int? editingIndex;

  @override
  void initState() {
    super.initState();
    _loadTasks();
  }

  Future<void> _loadTasks() async {
    final prefs = await SharedPreferences.getInstance();
    final savedTasks = prefs.getStringList('elder_tasks') ?? [];
    setState(() {
      tasks = savedTasks;
    });
  }

  Future<void> _saveTasks() async {
    final prefs = await SharedPreferences.getInstance();
    await prefs.setStringList('elder_tasks', tasks);
  }

  void _showTaskDialog({String? initialText, int? index}) {
```

```
    if (initialText != null) {
      _taskController.text = initialText;
      editingIndex = index;
    } else {
      _taskController.clear();
      editingIndex = null;
    }
    showDialog(
      context: context,
      builder: (context) {
        return AlertDialog(
          title: Text(editingIndex == null ? 'Add Reminder' : 'Edit
Reminder'),
          content: TextField(
            controller: _taskController,
            decoration: const InputDecoration(hintText: 'Enter reminder'),
            autofocus: true,
          ),
          actions: [
            TextButton(
              onPressed: () {
                Navigator.pop(context);
                _taskController.clear();
              },
              child: const Text('Cancel'),
            ),
            ElevatedButton(
              onPressed: () {
                final text = _taskController.text.trim();
                if (text.isNotEmpty) {
                  setState(() {
                    if (editingIndex == null) {
                      tasks.add(text);
                    } else {
                      tasks[editingIndex!] = text;
                    }
                  });
                  _saveTasks();
                  Navigator.pop(context);
                  _taskController.clear();
                }
              },
              child: const Text('Save'),
            ),
          ],
```

```
        );
      },
    );
  }

  void _deleteTask(int index) {
    setState(() {
      tasks.removeAt(index);
    });
    _saveTasks();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      drawer: const AppDrawer(),
      appBar: AppBar(title: const Text('Reminders')),
      body: tasks.isEmpty
          ? const Center(
              child: Text(
                'No reminders yet.\nTap + to add your first reminder.',
                textAlign: TextAlign.center,
                style: TextStyle(fontSize: 18),
              ),
            )
          : ListView.builder(
              itemCount: tasks.length,
              itemBuilder: (context, index) {
                return Card(
                  margin: const EdgeInsets.symmetric(
                    horizontal: 12,
                    vertical: 6,
                  ),
                  child: ListTile(
                    leading: const Icon(Icons.task_alt, color:
Colors.teal),
                    title: Text(tasks[index]),
                    trailing: Row(
                      mainAxisSize: MainAxisSize.min,
                      children: [
                        IconButton(
                          icon: const Icon(Icons.edit, color:
Colors.orange),
                          onPressed: () => _showTaskDialog(
                            initialText: tasks[index],
```

```
                            index: index,
                          ),
                        ),
                        IconButton(
                          icon: const Icon(Icons.delete, color:
Colors.red),
                          onPressed: () => _deleteTask(index),
                        ),
                      ],
                    ),
                  ),
                );
              },
            ),
        floatingActionButton: FloatingActionButton(
          onPressed: () => _showTaskDialog(),
          tooltip: 'Add Reminder',
          child: const Icon(Icons.add),
        ),
      );
  }
}

// Contact Admin screen with a form, input & mail launcher
class ContactAdminScreen extends StatefulWidget {
  const ContactAdminScreen({super.key});

  @override
  State<ContactAdminScreen> createState() => _ContactAdminScreenState();
}

class _ContactAdminScreenState extends State<ContactAdminScreen> {
  final _formKey = GlobalKey<FormState>();
  final _nameController = TextEditingController();
  final _emailController = TextEditingController();
  final _messageController = TextEditingController();

  Future<void> _sendEmail() async {
    final name = Uri.encodeComponent(_nameController.text.trim());
    final email = Uri.encodeComponent(_emailController.text.trim());
    final message = Uri.encodeComponent(_messageController.text.trim());

    final mailtoUrl =
        'mailto:caregiver@family.example?subject=ElderAid%20Contact&body=N
ame:%20$name%0AEmail:%20$email%0A%0AMessage:%20$message';
```

```
    if (await canLaunchUrl(Uri.parse(mailtoUrl))) {
      await launchUrl(Uri.parse(mailtoUrl));
    } else {
      ScaffoldMessenger.of(
        context,
      ).showSnackBar(const SnackBar(content: Text('Could not open mail
app')));
    }
  }

  @override
  void dispose() {
    _nameController.dispose();
    _emailController.dispose();
    _messageController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      drawer: const AppDrawer(),
      appBar: AppBar(title: const Text('Contact Caregiver / Admin')),
      body: Padding(
        padding: const EdgeInsets.all(16),
        child: Form(
          key: _formKey,
          child: ListView(
            children: [
              const Icon(Icons.contact_mail, size: 80, color:
Colors.teal),
              const SizedBox(height: 16),
              TextFormField(
                controller: _nameController,
                decoration: const InputDecoration(
                  labelText: 'Name',
                  border: OutlineInputBorder(),
                  prefixIcon: Icon(Icons.person),
                ),
                validator: (value) =>
                    value == null || value.isEmpty ? 'Enter your name' :
null,
              ),
              const SizedBox(height: 12),
```

```
            TextFormField(
              controller: _emailController,
              decoration: const InputDecoration(
                labelText: 'Email',
                border: OutlineInputBorder(),
                prefixIcon: Icon(Icons.email),
              ),
              keyboardType: TextInputType.emailAddress,
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'Enter your email';
                }
                final emailRegex = RegExp(r'^[^@]+@[^@]+\.[^@]+');
                if (!emailRegex.hasMatch(value)) {
                  return 'Enter valid email';
                }
                return null;
              },
            ),
            const SizedBox(height: 12),
            TextFormField(
              controller: _messageController,
              decoration: const InputDecoration(
                labelText: 'Message',
                border: OutlineInputBorder(),
                prefixIcon: Icon(Icons.message),
              ),
              maxLines: 5,
              validator: (value) => value == null || value.isEmpty
                  ? 'Enter your message'
                  : null,
            ),
            const SizedBox(height: 20),
            ElevatedButton.icon(
              icon: const Icon(Icons.send),
              label: const Text('Send'),
              onPressed: () {
                if (_formKey.currentState!.validate()) {
                  _sendEmail();
                }
              },
            ),
          ],
        ),
      ),
```

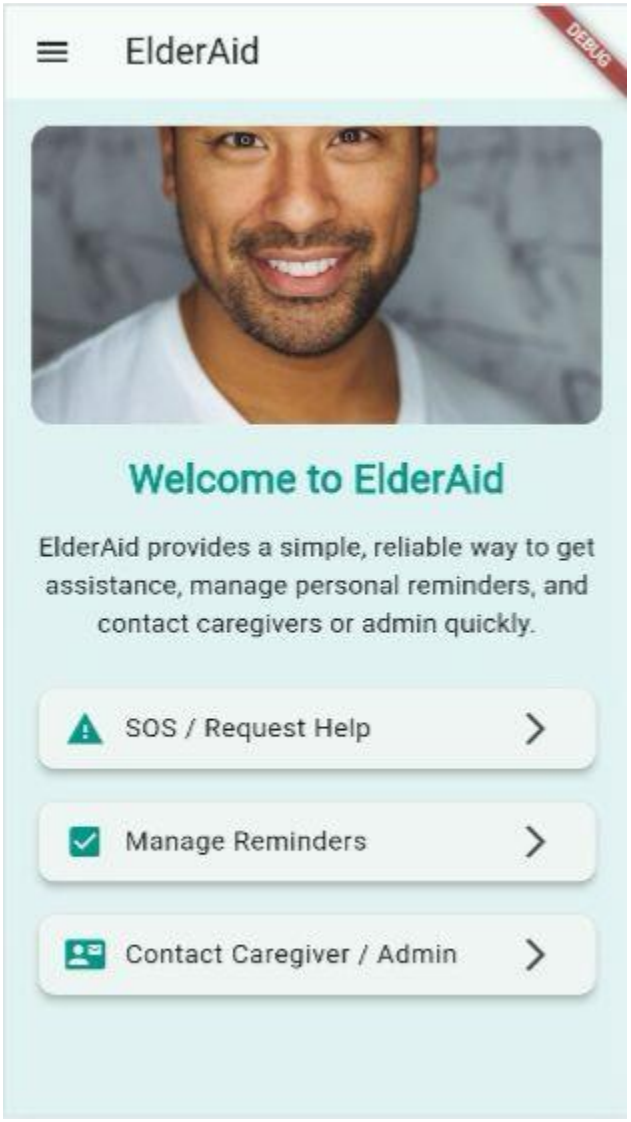
```
      ),
    );
  }
}

// About Screen
class AboutScreen extends StatelessWidget {
  const AboutScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      drawer: const AppDrawer(),
      appBar: AppBar(title: const Text('About')),
      body: Padding(
        padding: const EdgeInsets.all(16),
        child: ListView(
          children: const [
            Icon(Icons.favorite, size: 80, color: Colors.teal),
            SizedBox(height: 16),
            Text(
              'ElderAid',
              style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 12),
            Text(
              'ElderAid is a simple, easy-to-use companion app for elderly
users to request help quickly, manage reminders locally, and contact
caregivers or admin easily.',
              style: TextStyle(fontSize: 16),
            ),
            SizedBox(height: 24),
            Text(
              'Features:',
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.w600),
            ),
            SizedBox(height: 8),
            Text(
              '• Quick SOS actions: call, SMS, and share location via
email',
            ),
            Text('• Manage personal reminders saved locally'),
            Text('• Contact caregiver/admin through a prefilled mail
form'),
            Text('• Simple navigation with Drawer and Named Routes'),
```

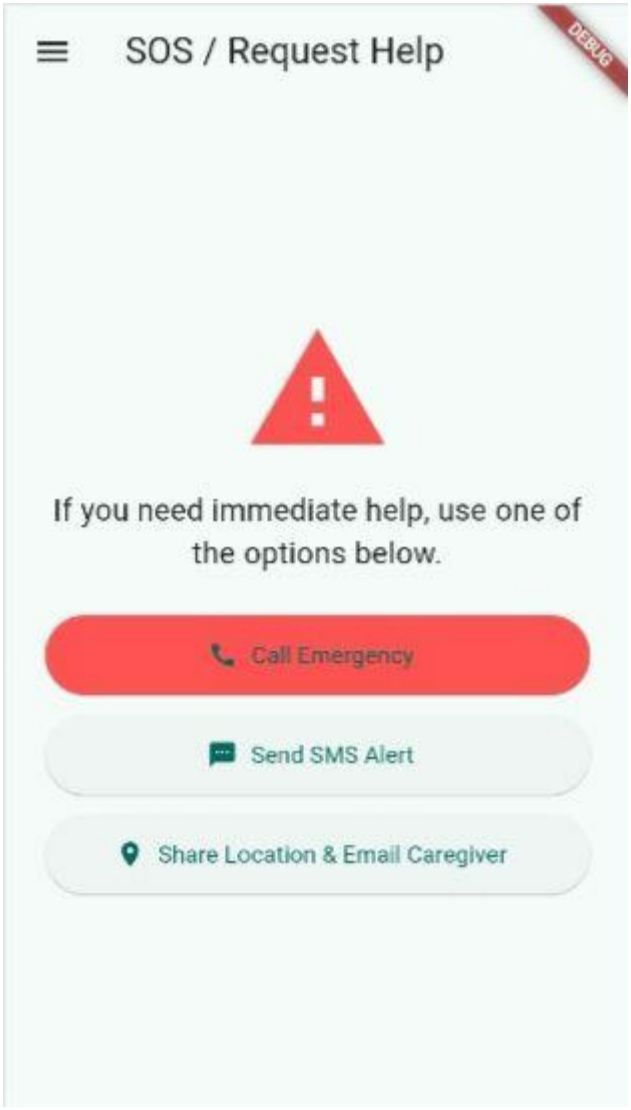
```
            SizedBox(height: 24),
            Text(
              'Developed as a sample mobile app using Flutter & Material
3.',
              style: TextStyle(fontSize: 14, fontStyle: FontStyle.italic),
            ),
          ],
        ),
      ),
    );
  }
}
```

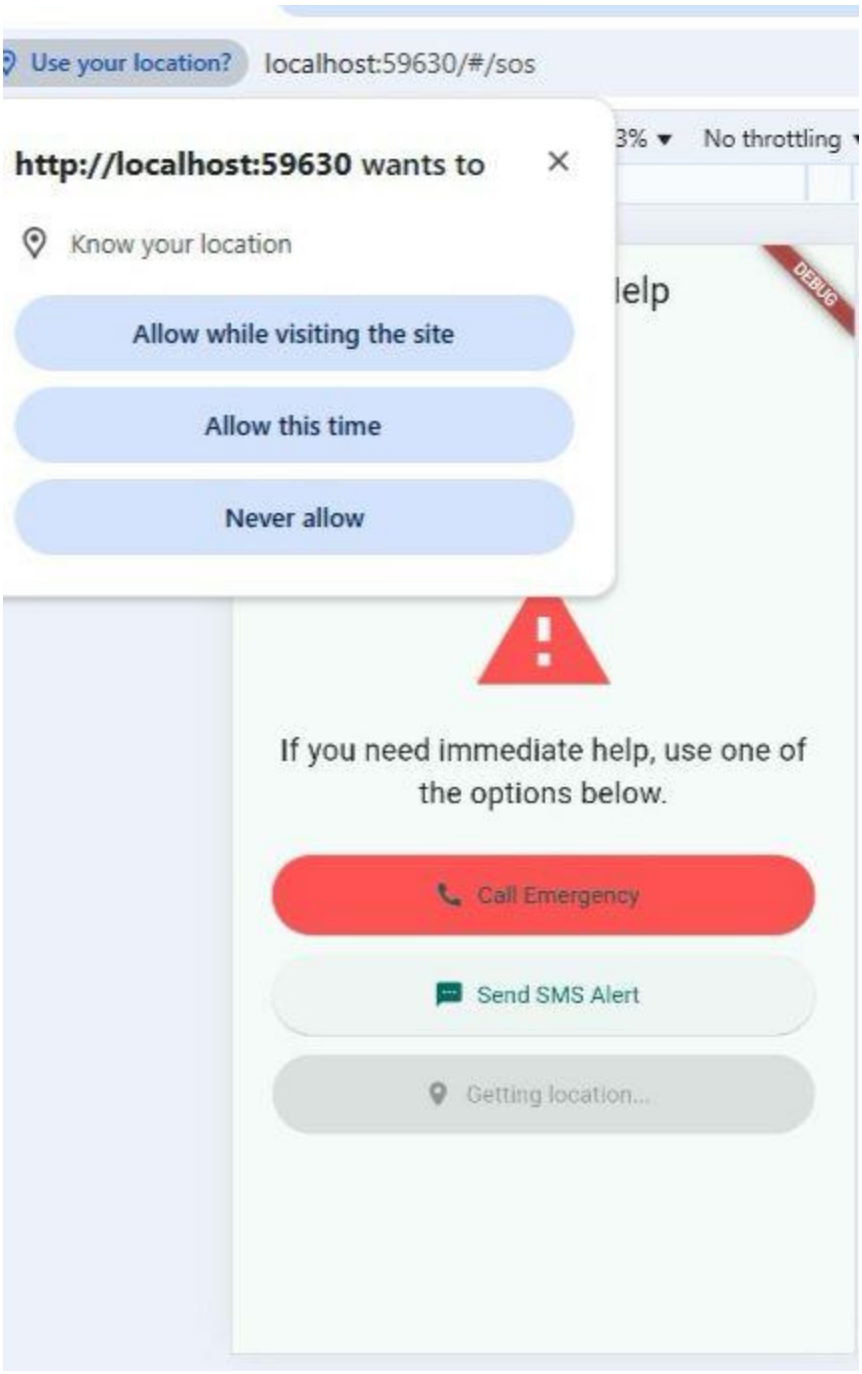
Pubspec.yaml

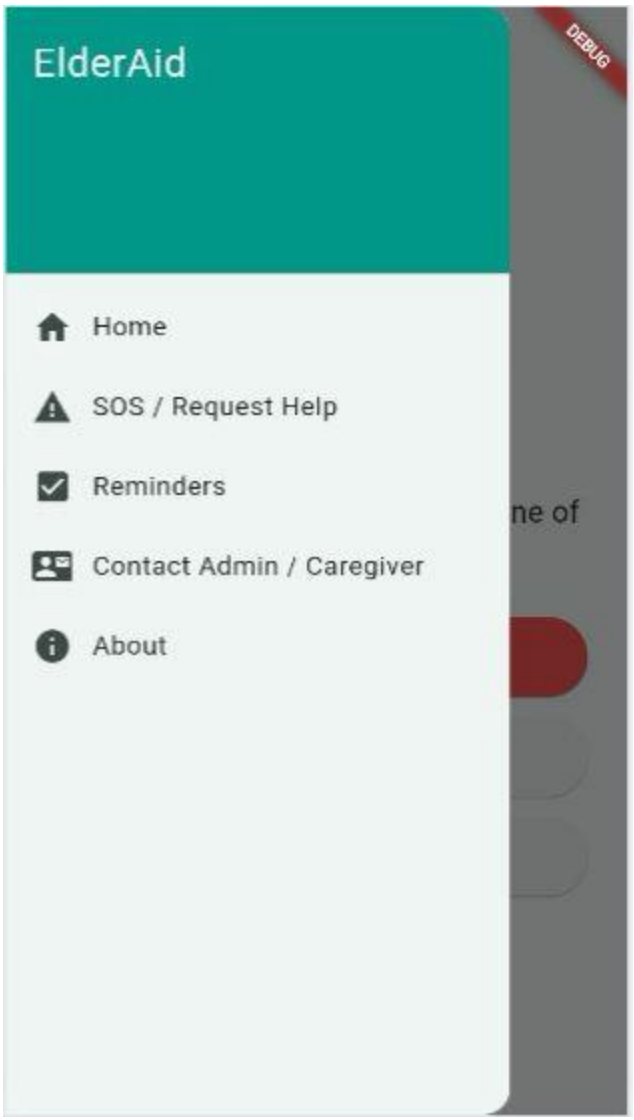
```
name: elderaid
description: A lifeline app for elderly users
publish_to: 'none'
version: 1.0.0+1

environment:
  sdk: ">=2.17.0 <4.0.0"

dependencies:
  flutter:
    sdk: flutter
  cupertino_icons: ^1.0.6
  shared_preferences: ^2.0.15
  url_launcher: ^6.1.7
  geolocator: ^9.0.2

dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^2.0.0

flutter:
  uses-material-design: true
```

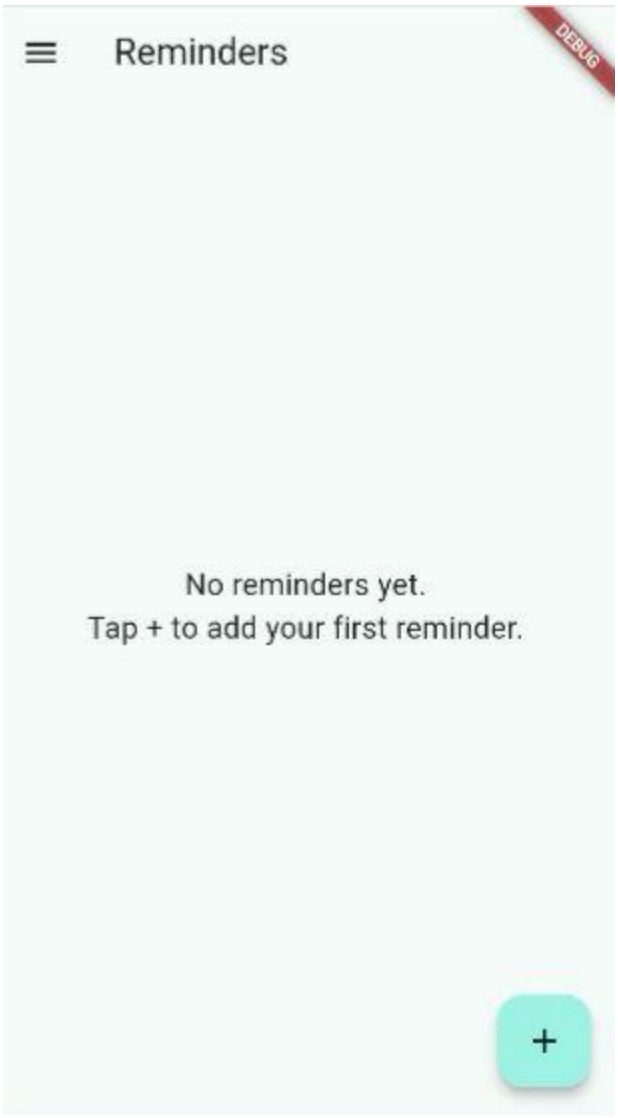
Result Snapshots

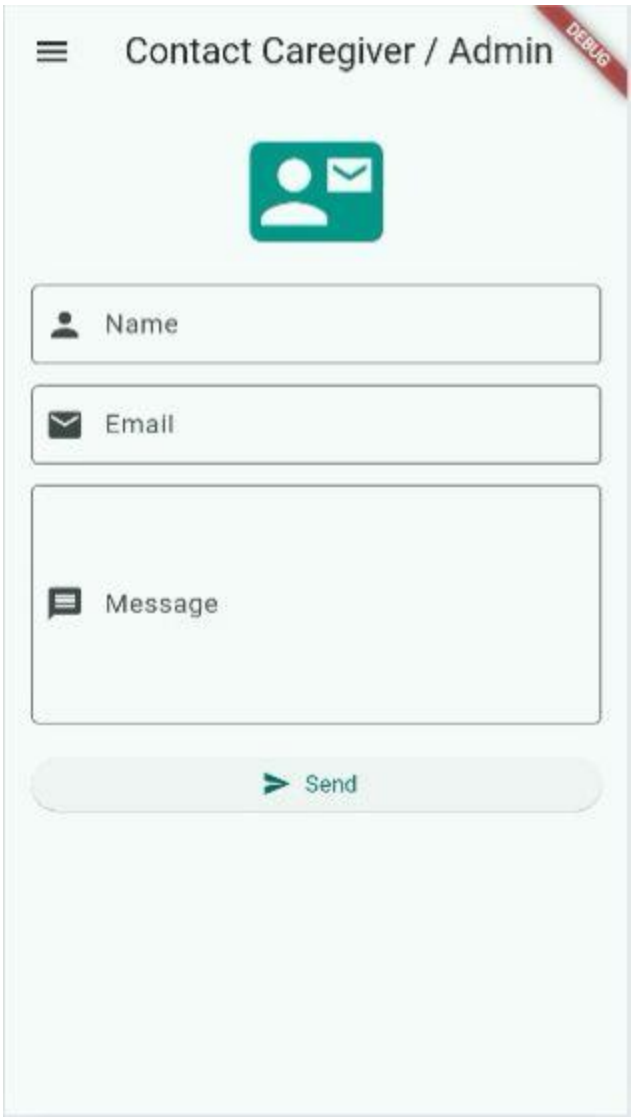


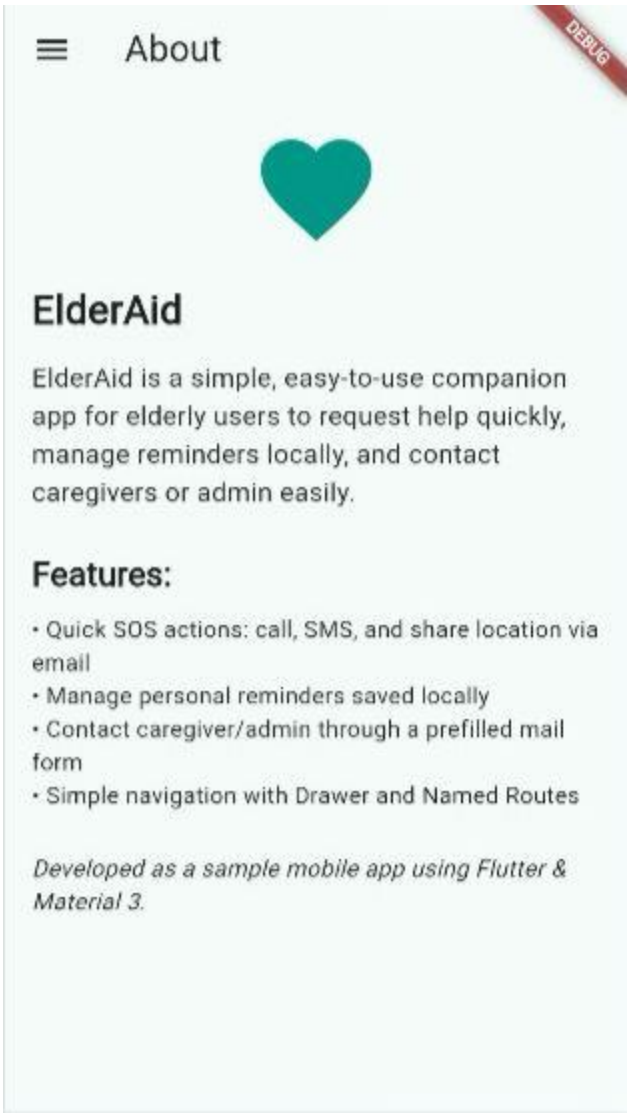

SOS / Request Help
DEBUG
If you need immediate help, use one of the options below.
Call Emergency
Send SMS Alert
Share Location & Email Caregiver

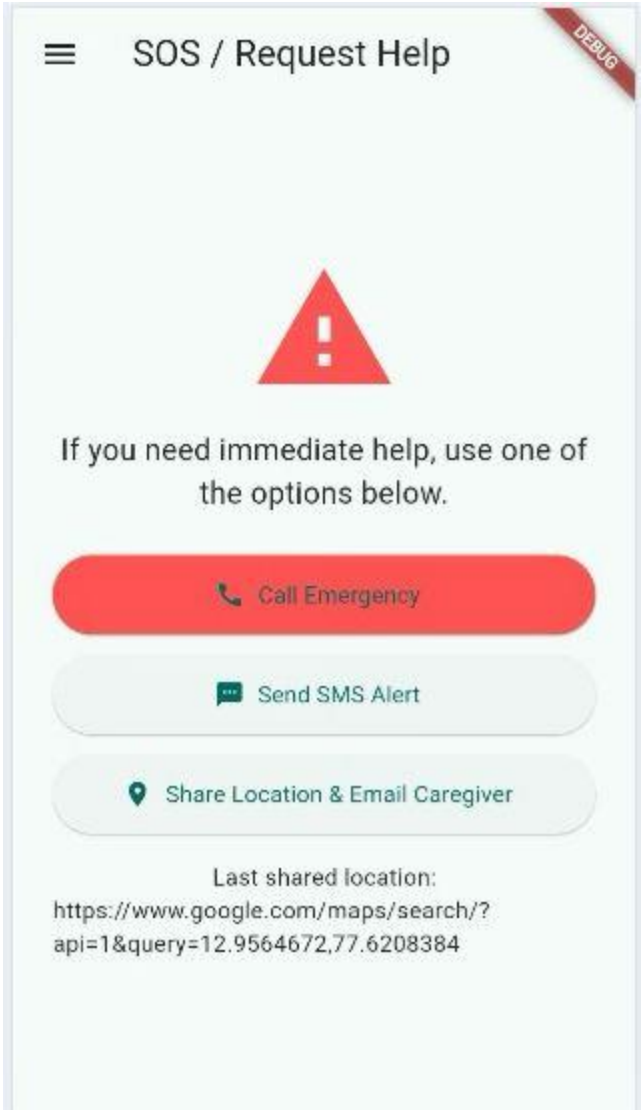

Use your location?
localhost:59630/#/sos
http://localhost:59630 wants to
Know your location
Allow while visiting the site
Allow this time
Never allow
3%
No throttling
lelp
DEBUG
If you need immediate help, use one of the options below.
Call Emergency
Send SMS Alert
Getting location...

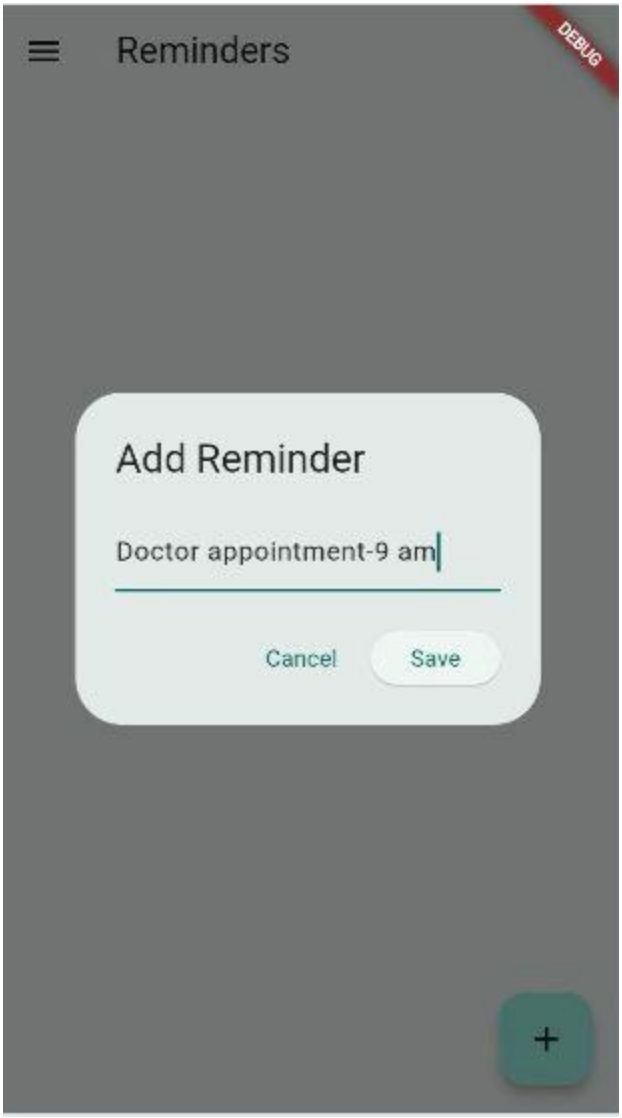

ElderAid
DEBUG
Home
SOS / Request Help
Reminders
Contact Admin / Caregiver
About
ne of

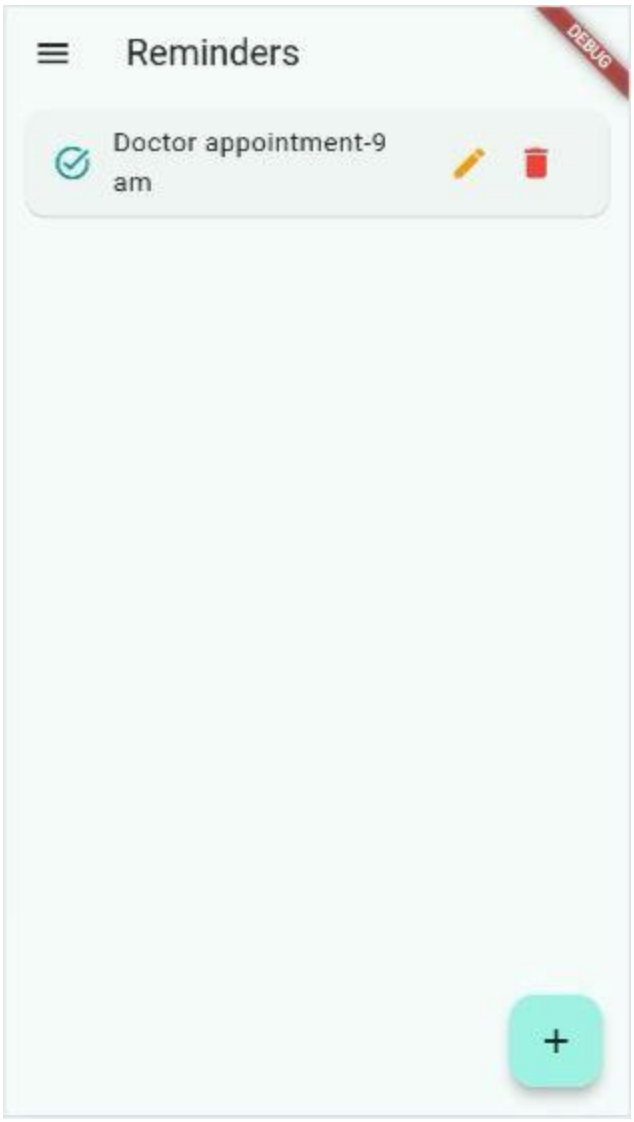

Reminders
DEBUG
No reminders yet.
Tap + to add your first reminder.
+


Contact Caregiver / Admin
DEBUG
Name
Email
Message
Send


About
DEBUG
ElderAid
ElderAid is a simple, easy-to-use companion app for elderly users to request help quickly, manage reminders locally, and contact caregivers or admin easily.
Features:
• Quick SOS actions: call, SMS, and share location via email
• Manage personal reminders saved locally
• Contact caregiver/admin through a prefilled mail form
• Simple navigation with Drawer and Named Routes
Developed as a sample mobile app using Flutter & Material 3.


SOS / Request Help
DEBUG
If you need immediate help, use one of the options below.
Call Emergency
Send SMS Alert
Share Location & Email Caregiver
Last shared location:
https://www.google.com/maps/search/?api=1&query=12.9564672,77.6208384


Reminders
DEBUG
Add Reminder
Doctor appointment-9 am
Cancel
Save
+


Reminders
DEBUG
Doctor appointment-9 am
+

Conclusion

This project demonstrated that building the Elder Aid app is not only about coding but also about thoughtful design, user accessibility, and real-world utility. By focusing on elders' needs and simplifying the process of requesting help, we created an application that is easy to use and reliable.

The app successfully enables elders to securely log in, submit assistance requests, and track their status in real-time. This improves their ability to seek timely help, especially during emergencies, and provides peace of mind to both users and their families.

The development process reinforced the importance of combining technical solutions with empathy and user-centered design. With a clean interface and streamlined features, Elder Aid can significantly enhance the independence and safety of elderly users.

Future expansions could include volunteer integration, live tracking, and additional accessibility features, making the app an even more comprehensive support tool. Overall, Elder Aid stands as a practical and impactful digital solution addressing the real challenges faced by elders in daily life.