SHEGuard

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

T.Yogitha	**(22501A05I3)**
R.B.Akshitha	**(22501A05F3)**
P.Sai Advaith	**(22501A05D6)**
P.Mithil	**(22501A05F2)**

Under the Guidance of

Mr. Michael Sadgun Rao Kona,

Assistant Professor

PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY

(Permanently affiliated to JNTU :: Kakinada, Approved by AICTE)

(An NBA & NAAC A+ accredited and ISO 9001:2015 Certified Institution)

Kanuru, Vijayawada - 520007

2024-25

PRASAD V POTLURI

SIDDHARTHA INSTITUTE OF TECHNOLOGY

(Permanently affiliated to JNTU :: Kakinada, Approved by AICTE)

(An NBA & NAAC A+ accredited and ISO 9001:2015 certified institution)

Kanuru, Vijayawada – 520007

CERTIFICATE

This is to certify that the project report title **"SHEGuard"** is the bonafide work of **T.Yogitha (22501A05I3) , R.B.Akshitha (22501A05F3) , P.Sai Advaith (22501A05D6) , P.Mithil (22501A05F2)** in partial fulfilment of completing the Academic project in Mobile App Development (20SA8651) during the academic year 2024-25.

Signature of the Incharge **Signature of the HOD**

INDEX

1. ABSTRACT

SHEGuard is more than just an app—it's a trusted safety companion designed to empower women with confidence and security. With a simple one-tap SOS button, users can instantly alert emergency contacts, nearby police, and safety networks when they need help the most. The app's live location tracking lets loved ones keep an eye on their journey, while automatic alerts in unsafe areas provide an extra layer of protection. Whether through voice or gesture-activated SOS triggers, SHEGuard ensures help is always within reach, even in the most critical moments. With end-to-end encryption safeguarding privacy and 24/7 support from verified responders and NGOs, SHEGuard isn't just about emergencies—it's about feeling safe, supported, and in control, no matter where life takes you.

Designed with an intuitive and user-friendly interface, SHEGuard ensures that women of all ages can easily access its features when needed. The app also includes community support, allowing users to connect with others in their area for added safety. AI-powered threat detection helps identify potential dangers before they escalate, giving users the confidence to move freely. Whether traveling alone, commuting at night, or simply exploring new places, SHEGuard provides peace of mind at every step. Safety is not a privilege—it's a right, and SHEGuard is here to uphold it.

1.1 SDG JUSTIFICATION REPORT

SDG Mapped: SDG 3 – Good Health and Well-Being, SDG 5 – Gender Equality, SDG 11 – Sustainable Cities and Communities

2.1 How SheGuard Supports SDG 3, SDG 5, and SDG 11

· **Enhanced Personal Safety (SDG 3, SDG 5, SDG 11):** SheGuard provides real-time safety alerts, emergency response features, and location tracking, ensuring women feel secure in public spaces and reducing risks to their well-being.

· **Empowerment Through Awareness (SDG 5):** The app includes educational resources on self-defense, legal rights, and safety guidelines, empowering women with knowledge and skills to protect themselves.

· **Community-Driven Safety Networks (SDG 5, SDG 11):** SheGuard fosters safe spaces by connecting users with local safety groups, authorities, and verified guardians, ensuring a supportive and responsive environment.

· **Efficient Emergency Management (SDG 3, SDG 11):** With features like instant SOS alerts, live tracking, and AI-driven threat detection, SheGuard enhances rapid emergency response, contributing to safer communities.

· **Scalability for Broader Impact (SDG 3, SDG 5, SDG 11):** SheGuard can be adopted across cities and institutions, integrating with public safety infrastructure to create safer urban environments.

2. INTRODUCTION

In today's fast-paced world, personal safety is a growing concern, especially for women who often navigate public spaces alone. **SHEGuard** is more than just a mobile application—it's a **powerful safety companion** designed to provide immediate assistance, real-time tracking, and proactive security measures. Whether commuting at night, traveling to unfamiliar places, or simply walking home alone, SHEGuard ensures that help is always within reach. With a **one-tap SOS button**, users can instantly alert **emergency contacts, nearby police stations, and registered safety networks**, ensuring a rapid response in times of distress.

Beyond emergency alerts, SHEGuard offers **live location tracking**, allowing trusted contacts to monitor movements in real time, providing an added layer of security and reassurance. The app is equipped with **automatic alerts that trigger when entering unsafe zones**, leveraging AI-driven technology to detect high-risk areas and notify users before potential threats arise. In situations where manual activation isn't possible, **voice and gesture-activated emergency triggers** ensure that users can discreetly call for help.

Privacy and security are at the core of SHEGuard, with **end-to-end encryption** protecting all user data, ensuring that personal information remains confidential. Additionally, SHEGuard connects users with a **network of verified responders, NGOs, and women's safety organizations**, offering **24/7 support** beyond emergencies. The app is designed with a **user-friendly interface**, making it accessible for women of all ages, backgrounds, and technical proficiencies.

At its heart, SHEGuard is not just about responding to threats—it's about **preventing them** and **empowering women to move through the world with confidence**. With a combination of cutting-edge safety features and community-driven support, SHEGuard redefines personal security, turning technology into a tool for independence and peace of mind. Because safety is not a privilege—it's a right, and SHEGuard is here to protect it.

3. OBJECTIVES AND SCOPE OF THE PROJECT

The primary objective of **SHEGuard** is to develop an efficient and user-friendly mobile application that enhances personal safety and provides real-time security assistance, particularly for women. The platform aims to:

1. **Ensure Rapid Emergency Response –** Implement a one-tap SOS feature that instantly alerts emergency contacts, local authorities, and nearby registered safety networks.

2. **Enable Real-Time Location Tracking –** Provide live GPS tracking that allows trusted contacts to monitor users' movements, ensuring continuous safety oversight.

3. **Integrate AI-Driven Safety Alerts –** Utilize AI technology to detect high-risk areas and automatically notify users when entering potentially unsafe zones.

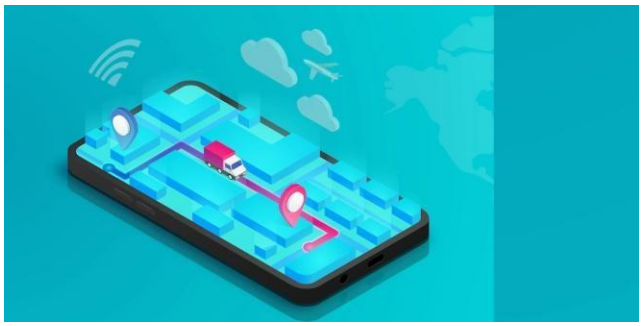
4. **Support Discreet Emergency Triggers** – Implement voice and gesture-based activation for emergency alerts, allowing users to call for help even in situations where manual activation isn’t possible.

5. **Ensure Data Privacy and Security** – Implement end-to-end encryption to protect user data, ensuring confidentiality and secure communication.

Scope

SHEGuard is designed to assist individuals—especially women—by providing proactive and responsive security features. The key areas of scope include:

- **User Registration & Authentication**: Secure login system allowing farmers to create profiles and manage their trusted contacts.
- **SOS Emergency Alerts:** A one-tap distress signal that notifies emergency contacts, police stations, and community responders.
- **Live Location Sharing:** Real-time GPS tracking that enables users to share their location with trusted contacts during commutes or high-risk situations.



- **Community & Support Integration:** A built-in network of verified responders, and safety organizations providing 24/7 assistance.
- **Scalability & Future Enhancements:** Future expansions may include self-defense tutorials, AI-powered threat detection, and integration with wearable safety devices for added protection.

4. SOFTWARE USED – EXPLANATION

The **SHEGuard** application is developed using a combination of **Android, Java, Firebase, Flask API, and various external APIs** to create a seamless and efficient personal safety platform.

Android

Android is an open-source mobile operating system developed by Google, designed primarily for touchscreen devices. It provides a robust platform for developing interactive, feature-rich mobile applications.

Usage in SHEGuard :

- The entire application is built on the Android platform, allowing users to access the security features from their smartphones.
- Android provides a flexible UI framework using XML for designing layouts, ensuring a user-friendly experience.
- The app utilizes Android components such as Activities, Fragments, RecyclerView, SearchView, and ProgressBar to create a seamless and responsive interface.

Java

Java is a high-level, object-oriented programming language widely used for Android app development due to its stability, security, and platform independence.

Usage in SHEGuard :

- Java is used to build the business logic of the app, handling user interactions, database connectivity, and API requests.
- It facilitates secure communication with Firebase, manages SOS triggers, and processes background location updates efficiently.
- Java's exception handling ensures smooth performance even in case of network failures or invalid user inputs.

Firebase

Firebase is a cloud-based backend platform by Google that provides real-time database management, authentication, and cloud storage for mobile applications.

Usage in SHEGuard

- Firebase Realtime Database is used to store and manage:
    - User profiles (registration, authentication, and contacts).

- Emergency alerts (real-time updates when SOS is triggered)..
- Live location tracking (for trusted contacts and responders).

- Firebase Authentication enables secure user login and registration, ensuring that only authenticated users can access personalized recommendations.
- Real-time synchronization allows emergency contacts to receive immediate notifications without requiring manual updates.

APIs (Application Programming Interfaces)

APIs allow applications to communicate with external services to fetch data dynamically.

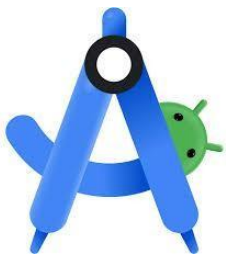
Usage in SHEGuard

- **Google Maps API:** Provides live location tracking and helps users share their location with emergency contacts.

- **Firebase API:** Used for storing and retrieving data efficiently in the cloud.
- **Custom Flask API:** Integrates AI-based risk assessment, analyzing user movement patterns and detecting high-risk areas.

Development Tools

Code Editor: Android Studio is the official IDE for Android development, providing a powerful code editor, visual layout editor, and debugging tools. It supports Java and integrates seamlessly with Firebase, making it an ideal choice for developing feature-rich mobile applications. The IDE includes an emulator for testing applications across different devices and screen sizes, ensuring a smooth user experience. Android Studio also offers real-time error detection and intelligent code completion, helping developers write optimized and bug-free code.



Version Control System (VCS): GitHub is a widely used version control system that helps manage project files, track changes, and collaborate with team members. It enables developers to work on different features simultaneously using branches and ensures that all modifications are safely stored and documented. By utilizing GitHub, the project remains organized, with a history of code changes that can be reviewed and reverted if needed. This improves code stability and allows multiple developers to contribute efficiently.

5. PROPOSED MODEL

1. Overview

The proposed SHEGuard system is a mobile-based personal safety application designed to provide real-time security assistance, emergency alerts, and proactive risk detection. The system enables users particularly women to instantly notify emergency contacts, track live locations, and receive AI-driven safety alerts in potentially dangerous situations. Additionally, it offers community-driven support, connecting users with verified responders and safety organizations to ensure a rapid response in emergencies.

2. System Components

User Authentication and Profile Management

Users can register and log in to access the system. The profile stores essential details such as name, email, contact number, emergency contacts, and trusted responders, ensuring quick communication during emergencies.

Real-Time Location Tracking Module

The system enables live location tracking, allowing trusted contacts to monitor the user's movements for added security. Google Maps API integration ensures accurate location sharing, updating contacts in real time.

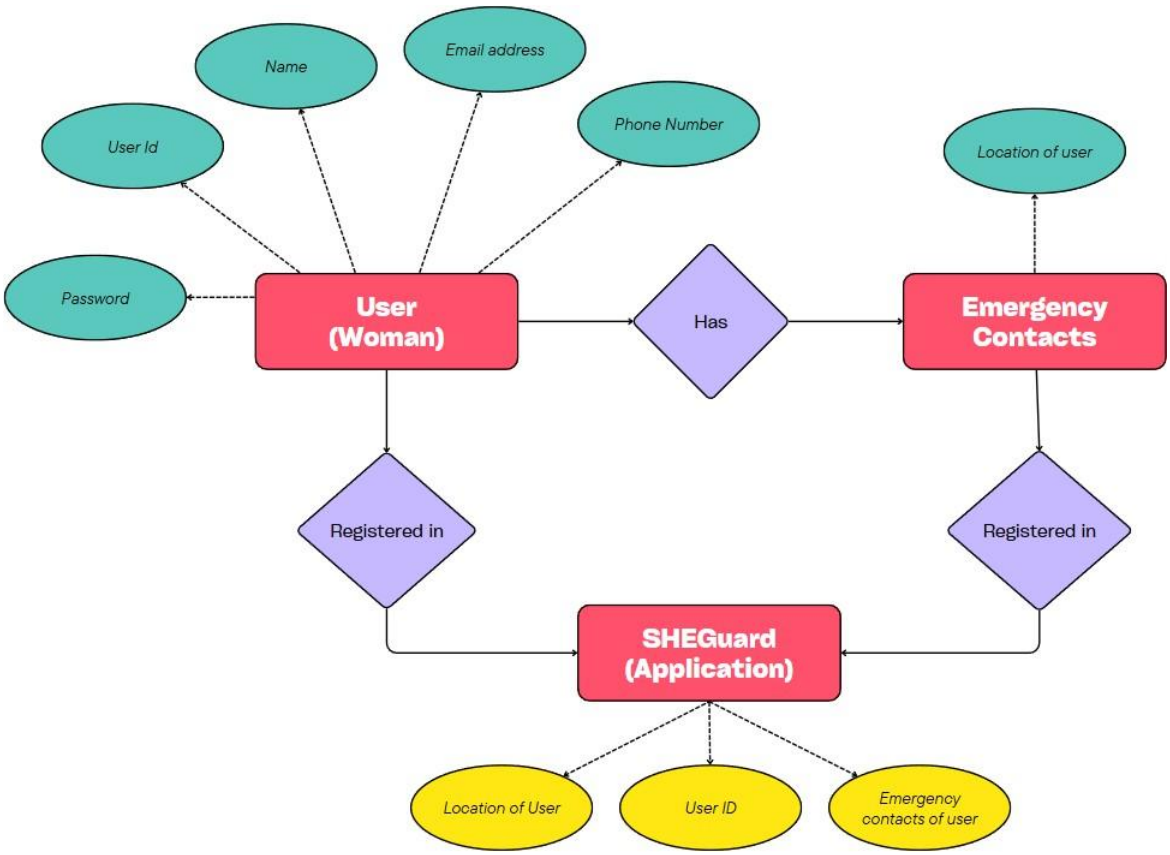
SOS Emergency Alert System

A one-tap SOS button allows users to instantly notify emergency contacts, police stations, and nearby safety responders. Gesture and voice-activated triggers ensure users can call for help discreetly, even if they are unable to access their phone manually.

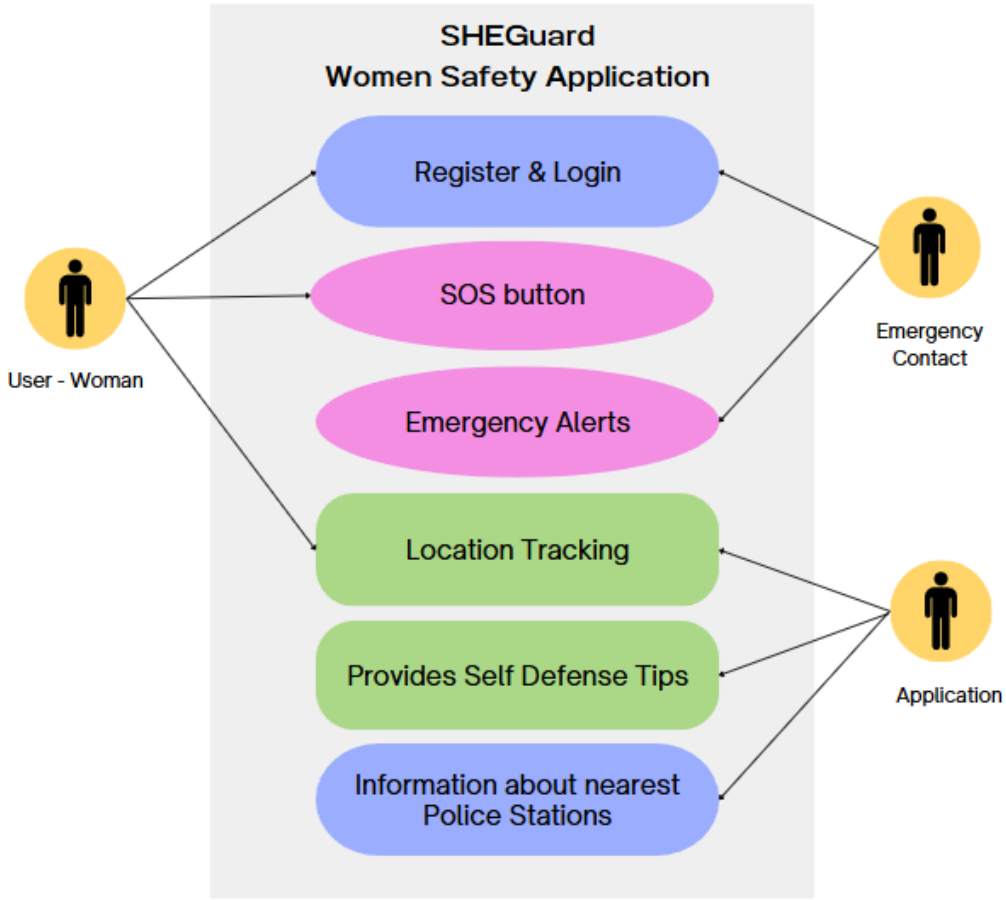
User Dashboard

The dashboard provides an overview of the user's details, including - Current location tracking details, Emergency contact list , Recent safety alerts and recommendations , Past SOS trigger history for reference and can manage trusted contacts, customize emergency settings, and review security insights from their profile.

3. Entity-Relationship Model



4. Use Case Model



6. SAMPLE CODE

Github repository link: https://github.com/yogithaaah/SheGuard

Folder Structure:

```
SheGuard/
├── .idea/
│   ├── inspectionProfiles/
│   ├── (various XML configuration files)
├── app/
│   ├── src/
│   │   ├── main/
│   │   │   ├── assets/
│   │   │   │   ├── police-operation-siren.mp3
│   │   │   ├── java/com/android/sheguard/
│   │   │   │   ├── api/
│   │   │   │   │   ├── MyFirebaseAuth.java
│   │   │   │   │   ├── NotificationAPI.java
│   │   │   │   ├── common/
│   │   │   │   │   ├── Constants.java
│   │   │   │   ├── config/
│   │   │   │   │   ├── Prefs.java
│   │   │   │   ├── model/
│   │   │   │   │   ├── ContactModel.java
│   │   │   │   │   ├── HelplineModel.java
│   │   │   │   │   ├── NotificationDataModel.java
│   │   │   │   │   ├── NotificationSenderModel.java
│   │   │   │   │   ├── NotificationTokenModel.java
│   │   │   │   │   ├── UserModel.java
│   │   │   │   ├── service/
│   │   │   │   │   ├── FireBaseMessageService.java
│   │   │   │   │   ├── SosService.java
│   │   │   │   ├── ui/
│   │   │   │   │   ├── activity/
│   │   │   │   │   │   ├── LoginRegisterActivity.java
│   │   │   │   │   │   ├── MainActivity.java
│   │   │   │   │   │   ├── OnBoardingActivity.java
│   │   │   │   │   ├── adapter/
│   │   │   │   │   ├── core/
│   │   │   │   │   ├── entity/
```

```
│   │   │   │   │   ├── fragment/
│   │   │   │   │   ├── view/
│   │   │   │   ├── util/
│   │   │   │   │   ├── AppUtil.java
│   │   │   │   │   ├── FirebaseUtil.java
│   │   │   │   │   ├── NotificationClient.java
│   │   │   │   │   ├── SosUtil.java
│   │   │   │   ├── SheGuard.java
│   │   │   ├── res/
│   │   │   │   ├── anim/
│   │   │   │   ├── color/
│   │   │   │   ├── drawable/
│   │   │   │   ├── layout/
│   │   │   │   ├── menu/
│   │   │   │   ├── mipmap/
│   │   │   │   ├── navigation/
│   │   │   │   ├── raw/
│   │   │   │   ├── transition/
│   │   │   │   ├── values/
│   │   │   │   ├── xml/
│   │   │   ├── AndroidManifest.xml
│   ├── build.gradle
│   ├── proguard-rules.pro
├── gradle/
│   ├── wrapper/
├── build.gradle
├── settings.gradle
├── gradlew
├── gradlew.bat
```

MainActivity.java

```
package com.android.sheguard.ui.activity;

import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.NavigationUI;

import com.android.sheguard.R;
import com.android.sheguard.databinding.ActivityMainBinding;
import com.android.sheguard.util.ObservableVariable;
import com.google.firebase.auth.FirebaseAuth;

@SuppressWarnings("FieldCanBeLocal")
public class MainActivity extends AppCompatActivity {

    public static ObservableVariable<Boolean> shakeDetection = new ObservableVariable<>();
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
    }

    public void toggleDrawer() {
        if (binding.drawerLayout.isDrawerOpen(binding.navView)) {
            binding.drawerLayout.closeDrawer(binding.navView);
        } else {
            binding.drawerLayout.openDrawer(binding.navView);
        }
    }

    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController(this,
R.id.fragmentContainerView);
```

```
        return NavigationUI.navigateUp(navController, binding.drawerLayout) ||
super.onSupportNavigateUp();
    }

    @Override
    protected void onStart() {
        super.onStart();

        FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
        if (firebaseAuth.getCurrentUser() == null || !firebaseAuth.getCurrentUser().isEmailVerified()) {
            startActivity(new Intent(MainActivity.this, OnBoardingActivity.class));
            finishAffinity();
        }
    }
}
```

LoginRegisterActivity.java

```
package com.android.sheguard.ui.activity;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import com.android.sheguard.R;
import com.android.sheguard.databinding.ActivityLoginRegisterBinding;
@SuppressWarnings("FieldCanBeLocal")
public class LoginRegisterActivity extends AppCompatActivity {
    private ActivityLoginRegisterBinding binding;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityLoginRegisterBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
    }
    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController(this,
R.id.fragmentContainerView);
        return navController.navigateUp() || super.onSupportNavigateUp();
    }
}
```

HomeFragment.java

```
package com.android.sheguard.ui.fragment;

import  android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.activity.result.ActivityResultCallback;
import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.contract.ActivityResultContracts;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.navigation.NavOptions;
import androidx.navigation.Navigation;

import com.android.sheguard.R;
import com.android.sheguard.common.Constants;
import com.android.sheguard.config.Prefs;
import com.android.sheguard.databinding.FragmentHomeBinding;
import com.android.sheguard.service.SosService;
import com.android.sheguard.ui.activity.LoginRegisterActivity;
import com.android.sheguard.ui.activity.MainActivity;
import com.android.sheguard.util.AppUtil;
import com.android.sheguard.util.FirebaseUtil;
import com.android.sheguard.util.SosUtil;
import com.google.android.material.navigation.NavigationView;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.Iterator;
import java.util.Map;
```

```
import java.util.Objects;

public class HomeFragment extends Fragment {

    private FragmentHomeBinding binding;

    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        binding = FragmentHomeBinding.inflate(inflater, container, false);
        View view = binding.getRoot();

        ((AppCompatActivity) requireActivity()).setSupportActionBar(binding.header.toolbar);
        binding.header.collapsingToolbar.setTitle(getString(R.string.activity_home_title));
        binding.header.collapsingToolbar.setSubtitle(getString(R.string.activity_home_desc,
getString(R.string.unknown_user)));
        setUserNameOnTitle();
        Objects.requireNonNull(((AppCompatActivity)
requireActivity()).getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
        Objects.requireNonNull(((AppCompatActivity)
requireActivity()).getSupportActionBar()).setHomeAsUpIndicator(R.drawable.ic_nav_drawer);

        NotificationManager notificationManager = (NotificationManager)
requireContext().getSystemService(Context.NOTIFICATION_SERVICE);
        NotificationChannel channel1 = new
NotificationChannel(getString(R.string.notification_channel_push),
getString(R.string.notification_channel_push), NotificationManager.IMPORTANCE_HIGH);
        NotificationChannel channel2 = new
NotificationChannel(getString(R.string.notification_channel_emergency),
getString(R.string.notification_channel_emergency),
NotificationManager.IMPORTANCE_DEFAULT);
        notificationManager.createNotificationChannel(channel1);
        notificationManager.createNotificationChannel(channel2);

        binding.sosButton.setOnClickListener(v -> {
            if (AppUtil.permissionsGranted(getContext()) && SosUtil.isGPSEnabled(requireContext()))
{
                SosUtil.activateInstantSosMode(requireContext());
            } else if (!AppUtil.permissionsGranted(getContext())) {
                multiplePermissions.launch(AppUtil.REQUIRED_PERMISSIONS);
            } else {
                SosUtil.turnOnGPS(requireContext());
            }
        });
```

```
MainActivity.shakeDetection.setValue(Prefs.getBoolean(Constants.SETTINGS_SHAKE_DETECTI
ON, false));
        MainActivity.shakeDetection.setOnChangeListener(newValue -> {
            binding.btnShakeDetection.setVisibility(newValue ? View.VISIBLE : View.GONE);
            updateButtonText();
            if (!newValue) {
                SosUtil.stopSosNotificationService(requireContext());
            }
        });

binding.btnShakeDetection.setVisibility(Prefs.getBoolean(Constants.SETTINGS_SHAKE_DETECT
ION, false) ? View.VISIBLE : View.GONE);


        updateButtonText();


        binding.btnShakeDetection.setOnClickListener(v -> {
            if (!SosService.isRunning) {
                if (AppUtil.permissionsGranted(getContext()) &&
SosUtil.isGPSEnabled(requireContext())) {
                    SosUtil.startSosNotificationService(requireContext());
                    Snackbar.make(requireActivity().findViewById(android.R.id.content),
getString(R.string.service_started), Snackbar.LENGTH_LONG).show();
                } else if (!AppUtil.permissionsGranted(getContext())) {
                    multiplePermissions.launch(AppUtil.REQUIRED_PERMISSIONS);
                } else {
                    SosUtil.turnOnGPS(requireContext());
                }
            } else {
                SosUtil.stopSosNotificationService(requireContext());
                Snackbar.make(requireActivity().findViewById(android.R.id.content),
getString(R.string.service_stopped), Snackbar.LENGTH_LONG).show();
            }


            updateButtonText();
        });


        binding.contacts.setOnClickListener(v ->
Navigation.findNavController(view).navigate(R.id.action_homeFragment_to_contactsFragment));
        binding.helpline.setOnClickListener(v ->
Navigation.findNavController(view).navigate(R.id.action_homeFragment_to_helplineFragment));
        binding.safetyTips.setOnClickListener(v ->
Navigation.findNavController(view).navigate(R.id.action_homeFragment_to_safetyTipsFragment));
        binding.about.setOnClickListener(v ->
Navigation.findNavController(view).navigate(R.id.action_homeFragment_to_aboutFragment));
```

```
        FirebaseUtil.updateToken();

        initializeDrawerItems();

        if (!AppUtil.permissionsGranted(getContext())) {
            multiplePermissions.launch(AppUtil.REQUIRED_PERMISSIONS);
        }

        return view;
    }

    private void initializeDrawerItems() {
        ((NavigationView)
requireActivity().findViewById(R.id.navView)).setNavigationItemSelectedListener(item -> {
            int id = item.getItemId();
            NavOptions navOptions = new NavOptions.Builder()
                    .setEnterAnim(0)
                    .setExitAnim(0)
                    .setPopEnterAnim(R.anim.slide_out)
                    .setPopExitAnim(R.anim.fade_in)
                    .build();

            if (id == R.id.nav_profile) {

Navigation.findNavController(binding.getRoot()).navigate(R.id.action_homeFragment_to_profileFra
gment, null, navOptions);
            } else if (id == R.id.nav_settings) {

Navigation.findNavController(binding.getRoot()).navigate(R.id.action_homeFragment_to_settingsFr
agment, null, navOptions);
            } else if (id == R.id.nav_logout) {
                FirebaseAuth.getInstance().signOut();
                Intent intent = new Intent(getContext(), LoginRegisterActivity.class);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                startActivity(intent);
            }

            ((MainActivity) requireActivity()).toggleDrawer();
            return true;
        });
    }
```

```
    public void setUserNameOnTitle() {
        final String[] userName = {getString(R.string.unknown_user)};

        FirebaseFirestore.getInstance()
                .collection(Constants.FIRESTORE_COLLECTION_USERLIST)
                .document(Objects.requireNonNull(FirebaseAuth.getInstance().getCurrentUser()).getUid())
                .get()
                .addOnCompleteListener(task -> {
                    if (task.isSuccessful()) {
                        DocumentSnapshot document = task.getResult();
                        if (document.exists()) {
                            userName[0] = document.getString("name");
                            Prefs.putString(Constants.PREFS_USER_NAME, userName[0]);
                        }
                    }

                    if (getContext() != null) {
                        binding.header.collapsingToolbar.setSubtitle(getString(R.string.activity_home_desc,
userName[0]));
                    }
                });
    }

    private void updateButtonText() {
        new Handler(Looper.getMainLooper()).postDelayed(() -> {
            if (getContext() != null) {
                binding.btnShakeDetection.setText(SosService.isRunning ?
getString(R.string.btn_stop_service) : getString(R.string.btn_start_service));
            }
        }, 200);
    }

    private final ActivityResultLauncher<String[]> multiplePermissions =
registerForActivityResult(new ActivityResultContracts.RequestMultiplePermissions(), new
ActivityResultCallback<Map<String, Boolean>>() {
        @Override
        public void onActivityResult(Map<String, Boolean> result) {
            Iterator<Map.Entry<String, Boolean>> it = result.entrySet().iterator();
            while (it.hasNext()) {
                Map.Entry<String, Boolean> pair = it.next();
                if (!pair.getValue()) {
                    Snackbar snackbar =
Snackbar.make(requireActivity().findViewById(android.R.id.content),
```

```
R.string.permission_must_be_granted, Snackbar.LENGTH_INDEFINITE);
                snackbar.setAction(R.string.grant, v -> {
                    multiplePermissions.launch(new String[]{pair.getKey()});
                    snackbar.dismiss();
                });
                snackbar.show();
            }

            if (!it.hasNext() && AppUtil.permissionsGranted(getActivity())) {
                binding.btnShakeDetection.performClick();
            }
        }
    }
});
}
```

ProfileFragment.java

```
package com.android.sheguard.ui.fragment;

import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Looper;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.Fragment;
import androidx.navigation.Navigation;

import com.android.sheguard.R;
import com.android.sheguard.common.Constants;
import com.android.sheguard.databinding.FragmentProfileBinding;
```

```
import com.android.sheguard.model.UserModel;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.Priority;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.io.IOException;
import java.util.List;
import java.util.Locale;
import java.util.Objects;

public class ProfileFragment extends Fragment {

    private FragmentProfileBinding binding;
    private LocationManager locationManager = null;
    private LocationRequest locationRequest = null;

    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        binding = FragmentProfileBinding.inflate(inflater, container, false);
        View view = binding.getRoot();

        ((AppCompatActivity) requireActivity()).setSupportActionBar(binding.header.toolbar);
        ActionBar actionBar = ((AppCompatActivity) requireActivity()).getSupportActionBar();
        if (actionBar != null) {
            actionBar.setDisplayHomeAsUpEnabled(true);
            actionBar.setDisplayShowHomeEnabled(true);
            binding.header.collapsingToolbar.setTitle(getString(R.string.activity_profile_title));
            binding.header.collapsingToolbar.setSubtitle(getString(R.string.activity_profile_desc));
        }

        if (locationRequest == null) {
            locationRequest = new LocationRequest.Builder(Priority.PRIORITY_HIGH_ACCURACY,
5000)
                    .setWaitForAccurateLocation(false)
                    .setMinUpdateIntervalMillis(2000)
                    .setMaxUpdateDelayMillis(5000)
```

```
                .build();
        }

        getUserDetails();
        getCurrentLocation();

        binding.btnEditProfile.setOnClickListener(v ->
Navigation.findNavController(view).navigate(R.id.action_profileFragment_to_editProfileFragment));

        return view;
    }

    private void getUserDetails() {
        FirebaseFirestore.getInstance()
                .collection(Constants.FIRESTORE_COLLECTION_USERLIST)
                .document(Objects.requireNonNull(FirebaseAuth.getInstance().getCurrentUser()).getUid())
                .get()
                .addOnCompleteListener(task -> {
                    if (task.isSuccessful()) {
                        DocumentSnapshot document = task.getResult();
                        if (document.exists()) {
                            UserModel user = new UserModel(
                                    document.getString("name"),
                                    document.getString("email"),
                                    document.getString("phone")
                            );

                            binding.tvName.setText(user.getName());
                            binding.tvEmail.setText(user.getEmail());
                            binding.tvPhone.setText(user.getPhone());
                        }
                    }
                });
    }

    private boolean isGPSEnabled() {
        if (locationManager == null) {
            locationManager = (LocationManager)
requireContext().getSystemService(Context.LOCATION_SERVICE);
        }

        return locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
```

```
    }

    private void getCurrentLocation() {
        if (!isGPSEnabled()) {
            binding.tvLocation.setText(R.string.gps_is_not_enabled);
            return;
        } else {
            binding.tvLocation.setText(R.string.getting_location);
        }

        if (ActivityCompat.checkSelfPermission(requireContext(),
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
&&
                ActivityCompat.checkSelfPermission(requireContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            return;
        }

        final int[] numberOfUpdates = {0};

        LocationServices.getFusedLocationProviderClient(requireContext())
                .requestLocationUpdates(locationRequest, new LocationCallback() {
                    @Override
                    public void onLocationResult(@NonNull LocationResult locationResult) {
                        super.onLocationResult(locationResult);
                        if (getContext() == null) {
                            return;
                        }

                        numberOfUpdates[0]++;

                        if (numberOfUpdates[0] >= 3) {
                            LocationServices.getFusedLocationProviderClient(getContext())
                                    .removeLocationUpdates(this);

                            if (locationResult.getLocations().size() > 0) {
                                int idx = locationResult.getLocations().size() - 1;
                                double latitude = locationResult.getLocations().get(idx).getLatitude();
                                double longitude = locationResult.getLocations().get(idx).getLongitude();

                                Geocoder geocoder;
                                List<Address> addresses;
```

```
                    geocoder = new Geocoder(getContext(), Locale.getDefault());

                    try {
                        addresses = geocoder.getFromLocation(latitude, longitude, 1);

                        if (addresses != null) {
                            StringBuilder address = new StringBuilder();

                            for (int i = 0; i <= addresses.get(0).getMaxAddressLineIndex(); i++) {
                                address.append(addresses.get(0).getAddressLine(i));

                                if (i < addresses.get(0).getMaxAddressLineIndex()) {
                                    address.append("\n");
                                }
                            }

                            binding.tvLocation.setText(address.toString());
                        } else {
                            binding.tvLocation.setText(getString(R.string.failed_to_get_location));
                        }
                    } catch (IOException e) {
                        binding.tvLocation.setText(getString(R.string.failed_to_get_location));
                        e.printStackTrace();
                    }
                }
            }
        }, Looper.getMainLooper());
    }
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawerLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:clipChildren="false"
    android:fillViewport="true"
```

```
    android:fitsSystemWindows="false"
    tools:context=".ui.activity.MainActivity">

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragmentContainerView"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:navGraph="@navigation/nav_home" />

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/navView"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        app:headerLayout="@layout/view_header_navigation_drawer"
        app:menu="@menu/nav_drawer" />

</androidx.drawerlayout.widget.DrawerLayout>
}
```

Fragment_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:clipChildren="false"
    android:fitsSystemWindows="true"
    tools:context=".ui.fragment.HomeFragment">

    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fillViewport="true"

app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">
```

```
<LinearLayout
    android:id="@+id/sos_button"
    android:layout_width="280dp"
    android:layout_height="280dp"
    android:layout_gravity="center"
    android:layout_marginVertical="28dp"
    android:background="@drawable/btn_large_sos"
    android:elevation="12dp"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:layout_width="168dp"
        android:layout_height="wrap_content"
        android:shadowColor="@color/shadow"
        android:shadowDx="-4"
        android:shadowDy="-4"
        android:shadowRadius="4"
        android:text="@string/sos"
        android:textAlignment="center"
        android:textColor="?attr/colorOnError"
        android:textSize="60sp" />

</LinearLayout>

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn_shake_detection"
    style="@style/Widget.Material3.Button.OutlinedButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="8dp"
    android:minHeight="@dimen/button_height"
    android:text="@string/btn_start_service"
    android:textSize="16sp"
    app:cornerRadius="@dimen/component_corner_radius" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="@dimen/margin_16"
    android:layout_marginTop="@dimen/margin_16"
    android:baselineAligned="false"
    android:orientation="horizontal">

    <LinearLayout
        android:id="@+id/contacts"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="@dimen/margin_4"
```

```
        android:layout_marginBottom="@dimen/margin_8"
        android:layout_weight="1"
        android:background="@drawable/container"
        android:baselineAligned="false"
        android:clickable="true"
        android:focusable="true"
        android:orientation="vertical"
        android:paddingVertical="15dp"
        android:paddingStart="20dp"
        android:paddingEnd="20dp">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/activity_contacts_title"
            android:textColor="?attr/colorOnSurface"
            android:textSize="16sp"
            android:textStyle="bold" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:text="@string/activity_contacts_desc"
            android:textColor="?attr/colorOnSurfaceVariant"
            android:textSize="15sp"
            tools:ignore="NestedWeights" />

        <ImageView
            android:layout_width="28dp"
            android:layout_height="28dp"
            android:layout_gravity="start|bottom"
            android:layout_marginTop="@dimen/margin_16"
            android:contentDescription="@android:string/untitled"
            android:src="@drawable/ic_contacts"
            app:tint="?attr/colorPrimary" />

    </LinearLayout>

    <LinearLayout
        android:id="@+id/helpline"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/margin_4"
        android:layout_marginBottom="@dimen/margin_8"
        android:layout_weight="1"
        android:background="@drawable/container"
        android:baselineAligned="false"
        android:clickable="true"
        android:focusable="true"
        android:orientation="vertical"
```

```
        android:paddingVertical="15dp"
        android:paddingStart="20dp"
        android:paddingEnd="20dp">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/activity_helpline_title"
            android:textColor="?attr/colorOnSurface"
            android:textSize="16sp"
            android:textStyle="bold" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:text="@string/activity_helpline_desc"
            android:textColor="?attr/colorOnSurfaceVariant"
            android:textSize="15sp"
            tools:ignore="NestedWeights" />

        <ImageView
            android:layout_width="28dp"
            android:layout_height="28dp"
            android:layout_gravity="start|bottom"
            android:layout_marginTop="@dimen/margin_16"
            android:contentDescription="@android:string/untitled"
            android:src="@drawable/ic_helpline"
            app:tint="?attr/colorPrimary" />

    </LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="@dimen/margin_16"
    android:baselineAligned="false"
    android:orientation="horizontal">

    <LinearLayout
        android:id="@+id/safety_tips"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginEnd="@dimen/margin_4"
        android:layout_marginBottom="@dimen/margin_8"
        android:layout_weight="1"
        android:background="@drawable/container"
        android:baselineAligned="false"
        android:clickable="true"
        android:focusable="true"
```

```
        android:orientation="vertical"
        android:paddingVertical="15dp"
        android:paddingStart="20dp"
        android:paddingEnd="20dp">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/activity_safety_tips_title"
            android:textColor="?attr/colorOnSurface"
            android:textSize="16sp"
            android:textStyle="bold" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:text="@string/activity_safety_tips_desc"
            android:textColor="?attr/colorOnSurfaceVariant"
            android:textSize="15sp"
            tools:ignore="NestedWeights" />

        <ImageView
            android:layout_width="28dp"
            android:layout_height="28dp"
            android:layout_gravity="start|bottom"
            android:layout_marginTop="@dimen/margin_16"
            android:contentDescription="@android:string/untitled"
            android:src="@drawable/ic_safety_tips"
            app:tint="?attr/colorPrimary" />

    </LinearLayout>

    <LinearLayout
        android:id="@+id/about"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/margin_4"
        android:layout_marginBottom="@dimen/margin_8"
        android:layout_weight="1"
        android:background="@drawable/container"
        android:baselineAligned="false"
        android:clickable="true"
        android:focusable="true"
        android:orientation="vertical"
        android:paddingVertical="15dp"
        android:paddingStart="20dp"
        android:paddingEnd="20dp">

        <TextView
            android:layout_width="match_parent"
```

```
            android:layout_height="wrap_content"
            android:text="@string/activity_about_title"
            android:textColor="?attr/colorOnSurface"
            android:textSize="16sp"
            android:textStyle="bold" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:text="@string/activity_about_desc"
            android:textColor="?attr/colorOnSurfaceVariant"
            android:textSize="15sp"
            tools:ignore="NestedWeights" />

        <ImageView
            android:layout_width="28dp"
            android:layout_height="28dp"
            android:layout_gravity="start|bottom"
            android:layout_marginTop="@dimen/margin_16"
            android:contentDescription="@android:string/untitled"
            android:src="@drawable/ic_about"
            app:tint="?attr/colorPrimary" />

      </LinearLayout>
    </LinearLayout>
  </LinearLayout>
</androidx.core.widget.NestedScrollView>

<include
  android:id="@+id/header"
  layout="@layout/view_header_expandable" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

7. RESULT/OUTPUT SCREEN SHOTS

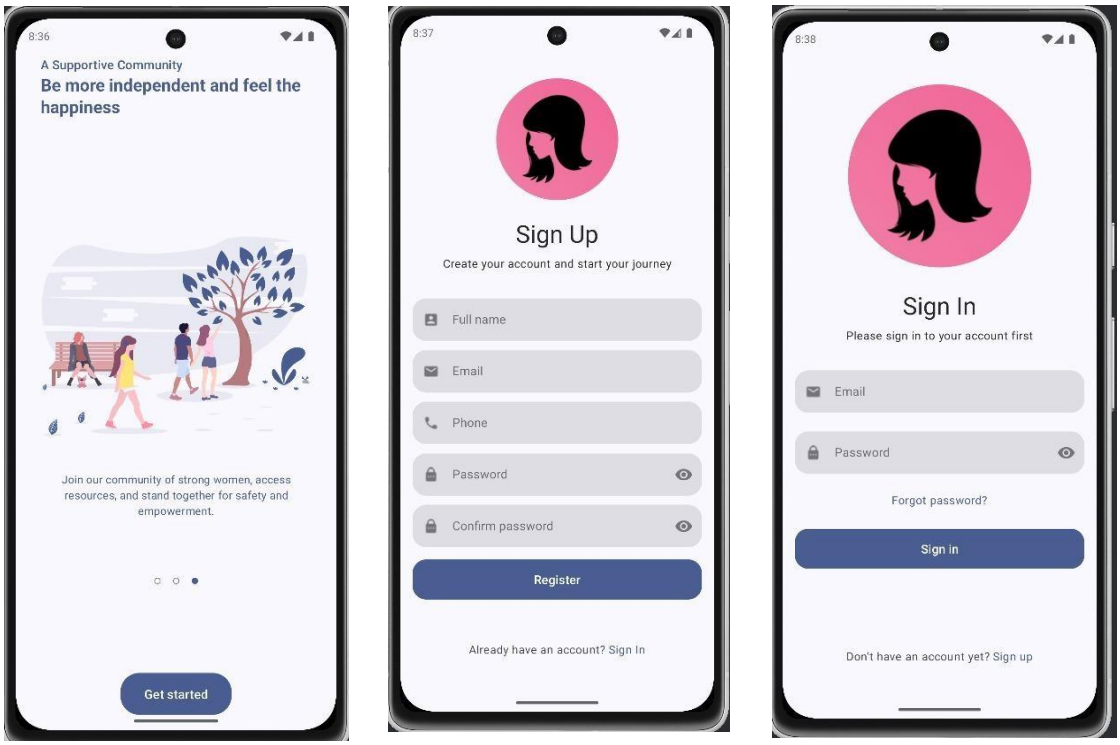


Figure 1: Sign-in and Sign-up page

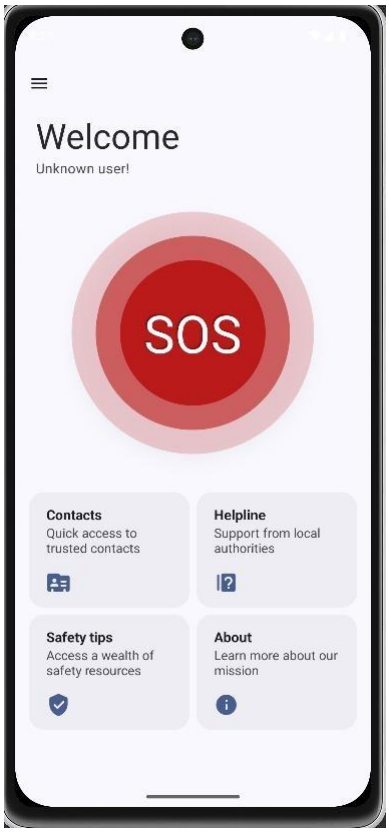


Figure 2: Home page

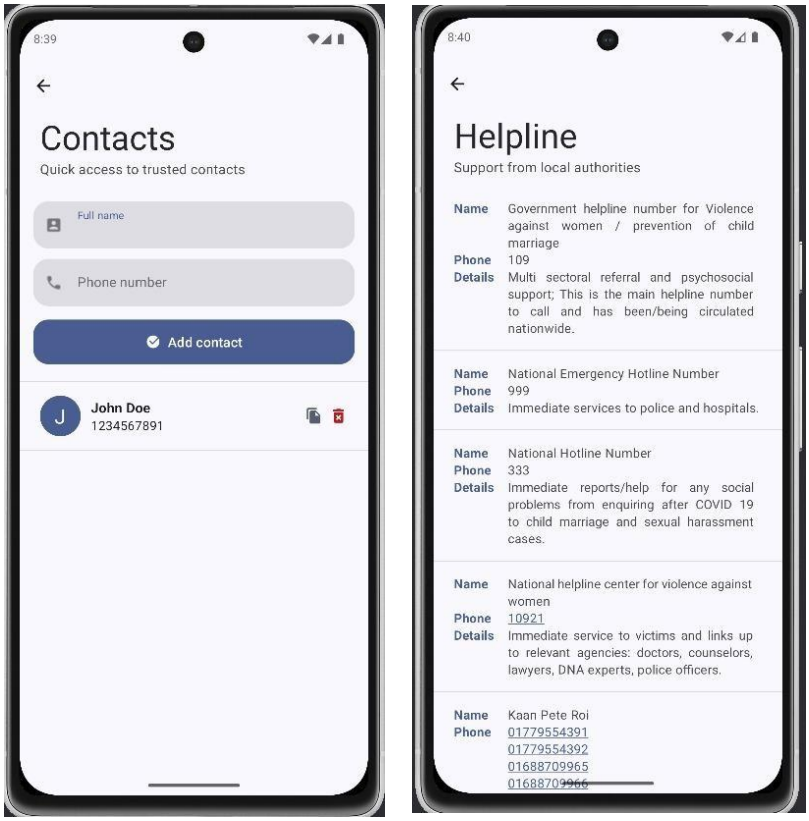


Figure 3: Contacts and Helpline Page

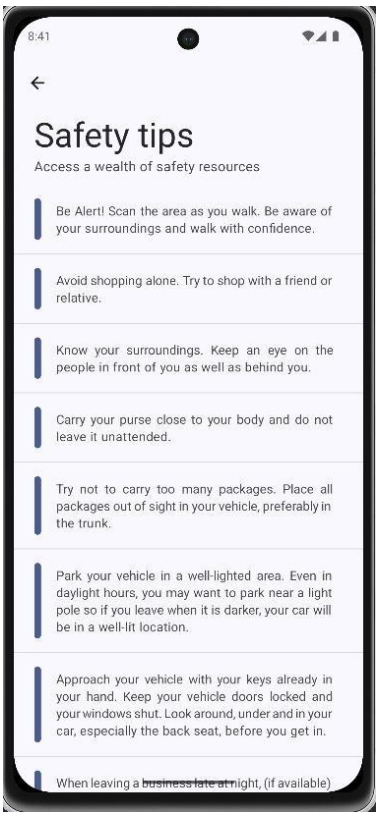


Figure 4: Safety tips Page

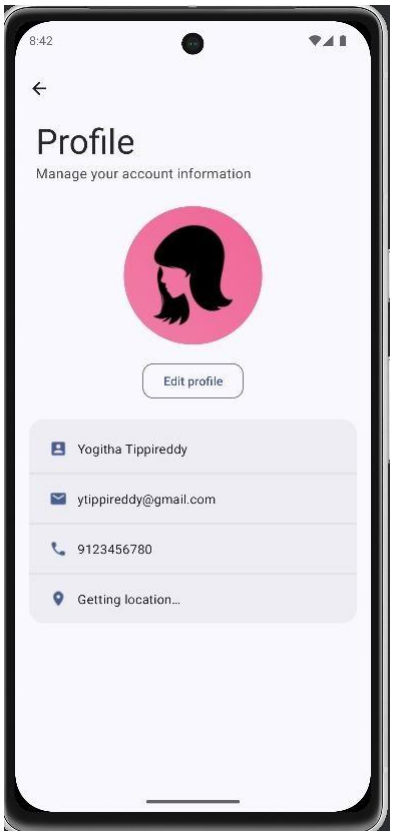


Figure 5: Profile Page

8. CONCLUSION

SHEGuard is not just an app—it's a powerful safety solution designed to protect, empower, and support women in any situation. In today's world, where personal security is a growing concern, SHEGuard bridges the gap by offering instant emergency assistance, real-time location tracking, and proactive safety features at the touch of a button. Whether facing an unexpected danger or simply wanting to feel safer while traveling, commuting, or walking alone, SHEGuard ensures that help is always just a tap away.

With its intelligent SOS system, automated alerts, and live tracking features, SHEGuard provides peace of mind to users and their loved ones. The app goes beyond just emergency responses—it also educates and prepares users through self-defense tips, preventive safety strategies, and expert guidance, making personal security more accessible than ever.

SHEGuard is built on the foundation of trust, privacy, and innovation. Its end-to-end encryption, customizable safety settings, and seamless user experience ensure that women can navigate the world with confidence. More than just a technological tool, SHEGuard represents a movement toward a safer and more secure future, where every woman can feel protected, independent, and empowered no matter where she goes. Because with SHEGuard, safety is not a privilege—it's a right.

9. REFERENCES (WEB SITE URLS)

GitHub repository link:

1. **Firebase Documentation:** https://firebase.google.com/docs
2. **Firebase Realtime Database:** https://firebase.google.com/docs/database
3. **Android Developer Guide:** https://developer.android.com/guide
4. **Android Studio:** https://developer.android.com/studio
5. **REST API with Flask:** https://realpython.com/flask-restful-api/