Aim: To include icons, images, fonts in Flutter app.

Theory:

Introduction

Flutter provides a rich set of widgets to create visually appealing user interfaces. Among these, icons, images, and custom fonts help improve the app's design and usability. Icons provide quick visual cues, images enhance user experience, and custom fonts allow unique styling to match the app's theme.

Implementation in Our Code

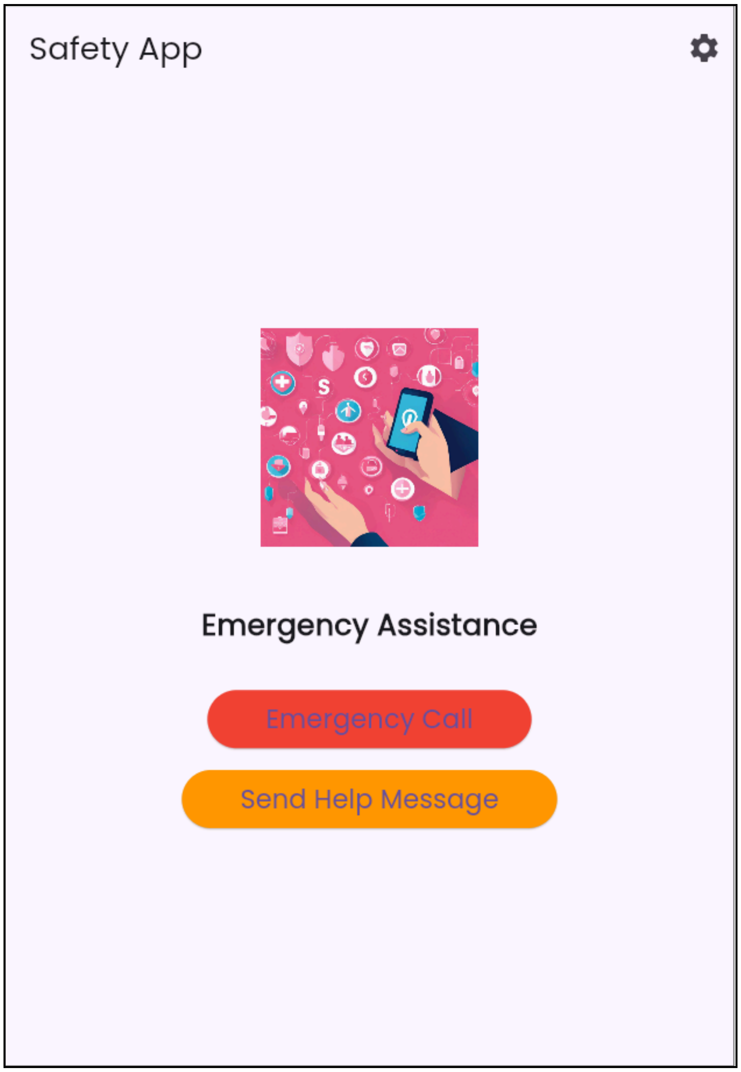
In this experiment, we added the following elements to our Woman Safety App:

1. Icons: Used Flutter's built-in Icons.security for the home screen and Icons.settings in the app bar.
2. Images: Added a safety-related image (safety_image.png) from the assets/ folder.
3. Fonts: Integrated the custom Google font Poppins for better typography.

Code :

```
import 'package:flutter/material.dart';

void main() {
  runApp(WomanSafetyApp());
}

class WomanSafetyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Woman Safety App',
      theme: ThemeData(
        primarySwatch: Colors.pink,
        fontFamily: 'Poppins', // Custom font
      ),
      home: HomeScreen(),
    );
  }
}

class HomeScreen extends StatelessWidget {
  void sendHelpMessage() {
    print("Help message sent!");
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Safety App'),
        actions: [
          IconButton(
            icon: Icon(Icons.settings),
            onPressed: () {
              Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => SettingsScreen()),
              );
            },
          ),
        ],
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Image.asset('assets/safety_image.png', width: 150), // Image
            SizedBox(height: 20),
            // Icon(Icons.security, size: 80, color: Colors.pink), // Icon
            SizedBox(height: 20),
            Text(
              "Emergency Assistance",
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 30),
            ElevatedButton(
              onPressed: () {
                print("Calling emergency number...");
              },
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.red,
                padding: EdgeInsets.symmetric(horizontal: 40, vertical: 15),
              ),
              child: Text("Emergency Call", style: TextStyle(fontSize: 18)),
            ),
            SizedBox(height: 15),
            ElevatedButton(
              onPressed: sendHelpMessage,
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.orange,
                padding: EdgeInsets.symmetric(horizontal: 40, vertical: 15),
              ),
              child: Text("Send Help Message", style: TextStyle(fontSize: 18)),
            ),
          ],
        ),
      ),
    );
  }
}

class SettingsScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Settings')),
      body: Center(
        child: Text(
          'Settings Page',
          style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
        ),
      ),
    );
  }
}
```

Screenshot:



Safety App

Emergency Assistance

Conclusion

In this experiment, we successfully added icons, images, and custom fonts to our Woman Safety App, enhancing its visual appeal and usability. Initially, we faced issues like missing asset configuration and font not applying, but we resolved them by correctly updating the pubspec.yaml file and ensuring assets were placed in the right directories.