SensorKit – Android Sensor Monitoring App

Abstract / Overview :

SensorKit is an Android application that demonstrates the integration and usage of mobile sensors. It reads data from sensors such as the **accelerometer, gyroscope, light, and proximity** sensors and displays real-time readings.

Purpose & Objectives:

- Provide an interactive platform for monitoring sensor data.
- Help students, developers, and researchers understand Android's sensors.

Target Audience:

- Engineering students learning Android development.
- Developers working on IoT/sensor-based applications.

Introduction :

Background: Modern smartphones include multiple sensors used in apps like fitness trackers, games, and navigation. However, a simple educational tool to demonstrate real-time sensor readings is often missing.

Need for the App: SensorKit provides a straightforward way to view live sensor data, making it useful for testing, learning, and prototyping.

Scope & Limitations:

- **Scope:** Demonstrates live readings of multiple sensors.
- **Limitations:** Requires supported sensors; limited offline functionality.

Objectives :

- Capture real-time sensor data using Android APIs.
- Display sensor values in a clean, intuitive UI.
- Provide modular structure for easy sensor extension.

Features

- Real-time monitoring of multiple sensors.
- Support for accelerometer, gyroscope, light, and proximity.
- Material Design–based UI.
- Modular, extendable code.

Technical Details

- **Platform & Tools:** Android Studio, Kotlin
- **Android Version:** Min SDK 21, Target SDK 34
- **Libraries:** Jetpack (MVVM), Material Design Components

Architecture

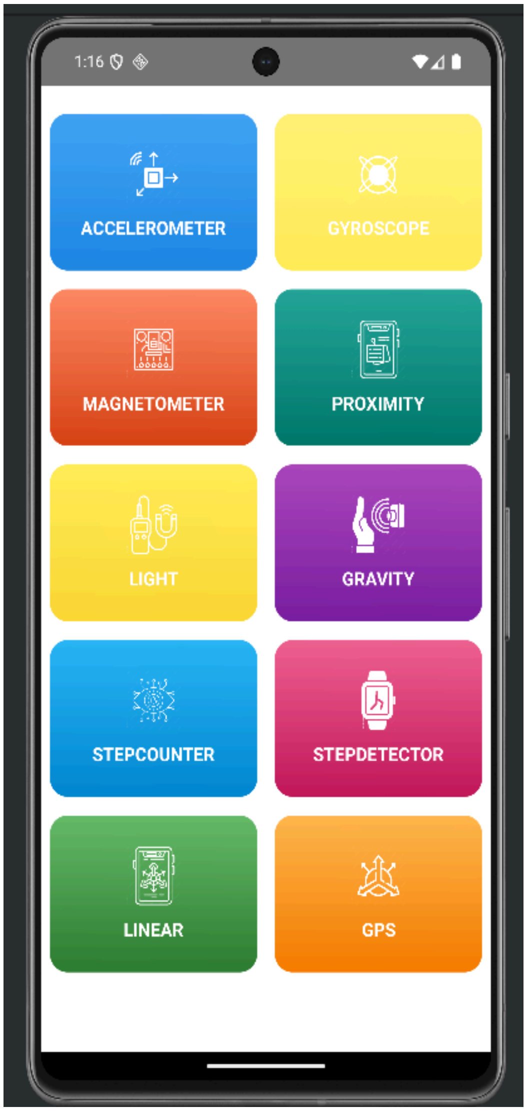
Pattern: MVVM (Model–View–ViewModel)

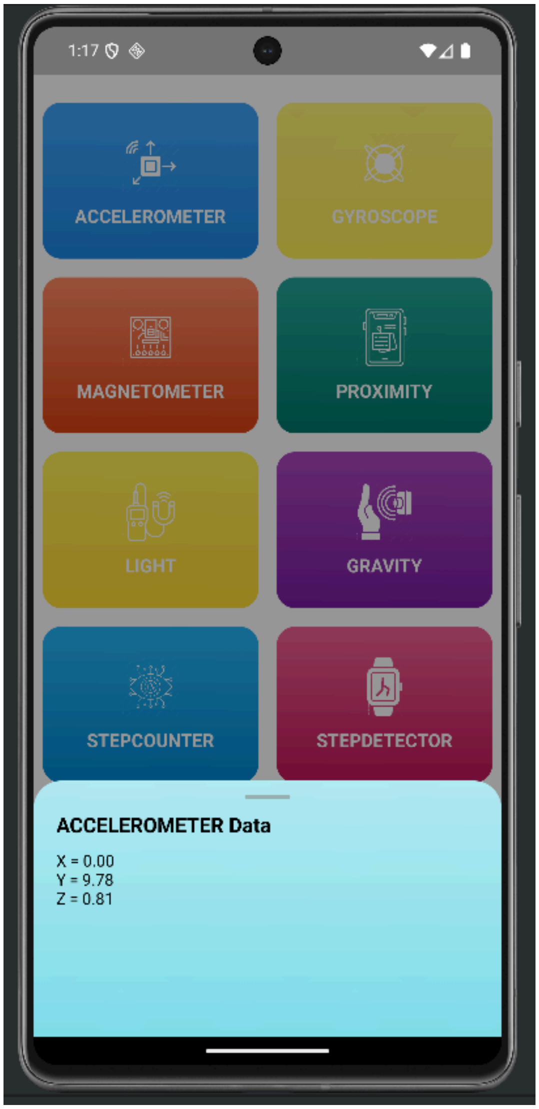
Flow:
UI Layer → ViewModel → Repository/Helper → Sensor Module

```
sensorkit/

├── MainActivity.kt            # Entry point of the app

|
├── sensors/                # Handles device sensor logic
│   ├── AccelerometerSensor.kt
│   ├── GyroscopeSensor.kt
│   ├── LightSensor.kt
│   └── ProximitySensor.kt
├── ui/                     # User Interface components
│   ├── HomeScreen.kt
│   ├── SensorDetailScreen.kt
│   └── adapters/
│       └── SensorAdapter.kt
├── utils/                  # Helper utilities & constants
│   ├── SensorUtils.kt
│   ├── Extensions.kt
│   └── Constants.kt
├── res/                    # Android resources (layouts, drawables, values)
│   ├── layout/
│   │   ├── activity_main.xml
│   │   ├── item_sensor.xml
│   │   └── sensor_detail.xml
│   └── drawable/
│       └── app_icon.png
├── AndroidManifest.xml     # Manifest file (permissions, activities)
├── build.gradle            # Module-level Gradle config
└── settings.gradle         # Project-level Gradle config
```

UI/UX Design

- **Screens:**





ACCELEROMETER Data

```
X = 0.00
Y = 9.78
Z = 0.81
```

- **Navigation:** Home → Sensor Details → Back to Home

Challenges & Solutions

- **Challenge:** Frequent updates caused UI lag
  Solution: Throttling with LiveData & coroutines
- **Challenge:** Missing sensors on some devices
  Solution: Availability checks before initialization

Future Enhancements

- Export data to CSV/Excel
- Graphical representation with charts
- Cloud sync & sharing
- Support for more sensors (GPS, barometer, heart rate)

Conclusion

SensorKit demonstrates Android’s sensor APIs in a **modular, user-friendly, and extendable** way. It can be a base for advanced sensor-based apps.

Lessons Learned:

- Applying MVVM architecture
- Handling device compatibility & performance