SLEEP TRACKING APP

Team Members Name	NM ID
SUNIL A B	ED8A92F5B230F1A938E4185D43966080
KAMALAKKANNAN P	C2ED0E44C4F9722BB894D96853026081
VIKNESH S	87FD63EA7F035ECA50641D9044D4BF73
VIJAY M	4FAE4D1C0C62F8EA2474C7338E163F81

Adding required Gradle scripts:

1. Project-level build.gradle

In the project-level build.gradle file, ensure you have the proper repositories and classpath for the Android Gradle Plugin.

gradle

Copy code

```
// Project-level build.gradle (root)
buildscript {
    repositories {
        google()
        mavenCentral()
    }
    dependencies {
        // Use the latest version of the Android Gradle plugin
        classpath 'com.android.tools.build:gradle:8.0.0' // Update as needed
        classpath 'org.jetbrains.kotlin:kotlin-gradle-plugin:1.8.0' // If using Kotlin
    }
}
```

```
allprojects {
    repositories {
        google()
        mavenCentral()
    }
}
```

2. Module-level build.gradle (App-level)

In your module-level build.gradle file, you need to define the necessary dependencies. These could include libraries for sensors, background tasks, UI components, and more.

gradle

```
// App-level build.gradle
plugins {
    id 'com.android.application'
    id 'kotlin-android' // If you're using Kotlin
}

android {
    compileSdkVersion 34 // Use the latest version of Android SDK
    defaultConfig {
        applicationId "com.example.sleeptracker" // Update with your app's ID
        minSdkVersion 21 // Minimum SDK for your app
        targetSdkVersion 34
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
```

```
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
        }
    }
}

dependencies {
    // AndroidX libraries
    implementation 'androidx.appcompat:appcompat:1.6.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'

    // For working with sensors (e.g., accelerometer, gyroscope, etc.)
    implementation 'androidx.core:core-ktx:1.10.1'
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.6.0'

    // For background tasks (important for tracking sleep data in the background)
    implementation 'androidx.work:work-runtime-ktx:2.8.0' // For background
processing

    // For graphs (optional, useful for displaying sleep data in a graphical format)
    implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0' //
MPAndroidChart for graphing

    // Permission handling (to access storage or sensors)
```

```
    implementation 'com.karumi:dexter:6.2.2' // Dexter for runtime permission requests


    // For using sensors (optional, based on your needs)
    implementation 'androidx.activity:activity-ktx:1.8.0' // For Activity/Fragment lifecycle management


    // Testing libraries (optional, if you're planning to write tests)
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'
}
```

3. Add Permissions in AndroidManifest.xml

Sleep tracking apps often require permissions like ACCESS_FINE_LOCATION, READ_EXTERNAL_STORAGE, and WRITE_EXTERNAL_STORAGE (depending on your app's features), as well as sensors access. Add the following permissions to your AndroidManifest.xml file:

xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sleeptracker">


    <application
        android:allowBackup="true"
        android:label="Sleep Tracker"
        android:theme="@style/Theme.SleepTracker">


        <!-- Required permissions for background services and storage -->
        <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

```
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission
android:name="android.permission.ACTIVITY_RECOGNITION"/>


    <!-- Permissions for working with sensors (accelerometer, gyroscope, etc.)
-->
    <uses-permission android:name="android.permission.BODY_SENSORS"
/>


    <!-- Optional: For background work with WorkManager -->
    <uses-permission
android:name="android.permission.ACCESS_BACKGROUND_LOCATION"/
>


    <!-- For using internet (if you need to send data to the server, sync data,
etc.) -->
    <uses-permission android:name="android.permission.INTERNET"/>


  </application>
</manifest>
```

4. Optional: Set Up Background Tracking with WorkManager

Since sleep tracking often requires background services, you might want to use WorkManager for handling background tasks such as tracking sleep data while the app isn't actively running. You can define a Worker to track sleep patterns in the background.

kotlin

```
import androidx.work.Worker
```

```
import androidx.work.WorkerParameters


class SleepTrackingWorker(appContext: Context, workerParams:
WorkerParameters) : Worker(appContext, workerParams) {


    override fun doWork(): Result {
        // Your code to track sleep data here
        // For example, check if the user is sleeping using sensors


        // Return success after completing the task
        return Result.success()
    }
}
```

To schedule the worker, use WorkManager:

kotlin

```
import androidx.work.OneTimeWorkRequest
import androidx.work.WorkManager


val sleepTrackingRequest =
OneTimeWorkRequest.Builder(SleepTrackingWorker::class.java)
    .build()


WorkManager.getInstance(context).enqueue(sleepTrackingRequest)
```

5. Sync Gradle

After making these changes to your build.gradle files, click Sync Now in the top bar of Android Studio or use the File > Sync Project with Gradle Files option to sync the project and download the necessary dependencies.

Screenshots: