A Sleep Tracking App For a Better Night's Rest

Project Based Experiential Learning Program

COLLEGE NAME :

BONAM VENKATA CHALAMAYYA INSTITUTE OF TECHNOLOGY AND SCIENCES ,AMALAPURAM

STUDENTS DETAILS :

Guthula. UdayBhaskar

GUTHULAUDAY1174@GMAIL.COM

REG: 21H45A0309

GUTTULA. SRIRAMAKRISHNA

GSRK1430@GMAIL.COM

REG: 20H41A0481

VAKAPALLI. Charan Sri Sai Teja

VAKAPALLICHARAN169@GMAIL.COM

REG: 20H41A05C5

AMBATI. NAGENDRABABU

AMBATINAGENDRA18@GMAIL.COM

REG: 20H41A0407

A Sleep Tracking App For a Better Night's Rest

1.Introduction

Overview

A project that demonstrates the use of Android Jetpack Compose to build a UI for a sleep tracking app. The app allows users to track their sleep.With the “Sleep Tracker” app, you can assess the quality of

sleep they have had in a day.

When most people think about getting a good night’s rest, they think about the length of time they sleep. But so much more goes into a night of good-quality sleep than just eight straight hours.

Helping you understand the true quality of your sleep is where sleep trackers come in.

Sleep trackers keep tabs on a variety of biological metrics, delivering invaluable data that can help you

better understand your sleep patterns, challenges, and areas of potential improvement — all of which you can use to get better, deeper rest, if you understand what you’re seeing.

“Sleep metrics can give people a better understanding of their sleep patterns and peculiarities,” says Dr. Po-Chang Hsu, who specializes in neurology, pediatrics,

and neonatology. “This info can help them identify any sleep issues and what lifestyle habits or underlying conditions might be causing them. Consequently, sleep metrics can help people improve their sleep by helping them change their habits, adopt new sleep hygiene, or tackle any other underlying issues.”

But how, exactly, do you use those metrics to get bettersleep?

Here are some of the key things your sleep tracker is telling you — and how to use those insights to improve your sleep quality and get more (and better!) rest:

When most people think about getting a good night's rest, they think about the length of time they sleep. But so much more goes into a night of good-quality sleep than just eight straight hours.

Helping you understand the true quality of your sleep is where sleep trackers come in.

Sleep trackers keep tabs on a variety of biological metrics, delivering invaluable data that can help you

better understand your sleep patterns, challenges, and areas of potential improvement — all of which you can use to get better, deeper rest, if you understand what you're seeing.

"Sleep metrics can give people a better understanding of their sleep patterns and peculiarities," says Dr. Po-Chang Hsu, who specializes in neurology, pediatrics,

and neonatology. "This info can help them identify any sleep issues and what lifestyle habits or underlying conditions might be causing them. Consequently, sleep metrics can help people improve their sleep by helping them change their habits, adopt new sleep hygiene, or tackle any other underlying issues."

But how, exactly, do you use those metrics to get bettersleep?

Here are some of the key things your sleep tracker is telling you — and how to use those insights to improve your sleep quality and get more (and better!) rest:

PURPOSE :

Fall asleep faster – Many sleep apps offer soothing sound effects, white noise and bedtime activities to help you fall asleep faster.

Improve sleep cycle – Other sleep apps track data points while users sleep to help them betterunderstand their sleep patterns.

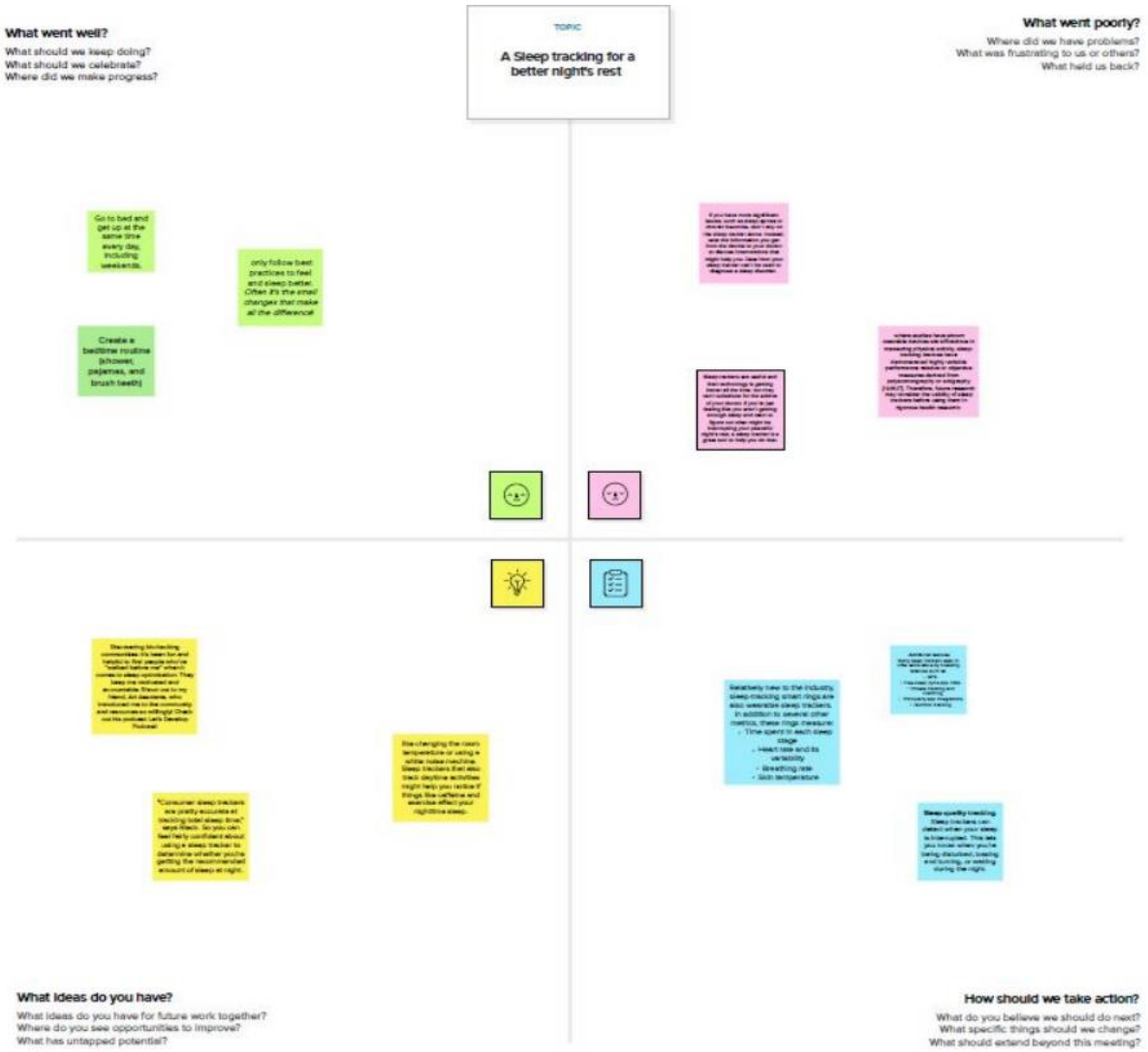
Develop healthy sleep habits – Some sleep apps provide educational resources to help users learn to improve their overall sleep habits.

Boost health and wellness – Lack of sleep can carry a whole host of health problems, both physically and mentally. Better health and wellness, therefore, is just one of the positive benefits of a great night of sleep

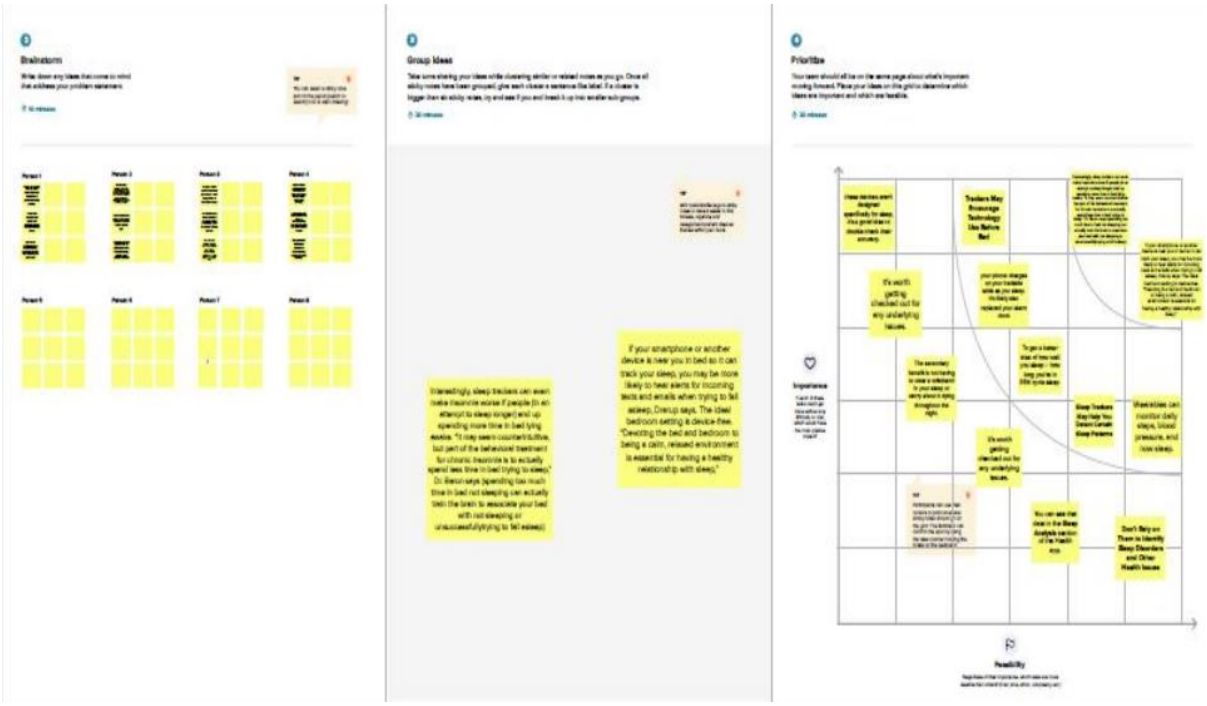
Sleep tracking considered by some to cause only stress and anxiety, where others find it life changing for the benefit of better sleep

2.PROBLEM DEFINITION & DESIGN THINKING

EMAPTHY MAP

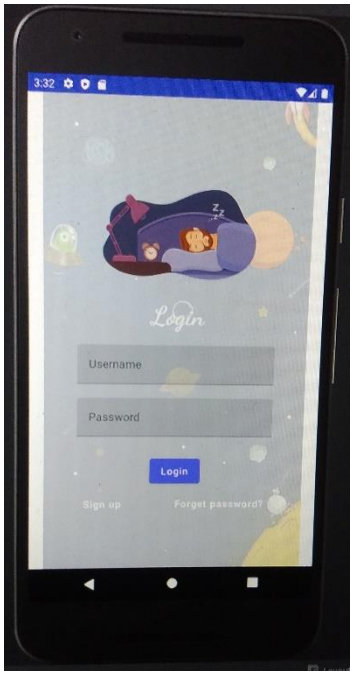


IDEATION AND BRAINSTROM

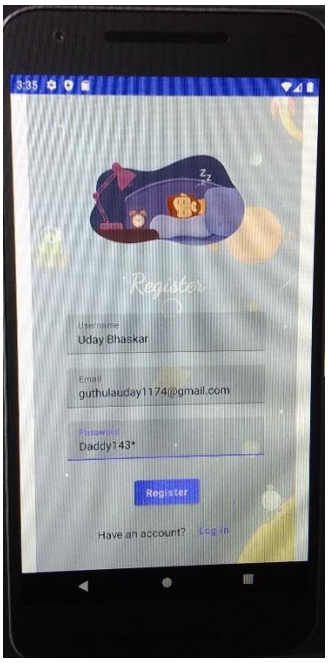


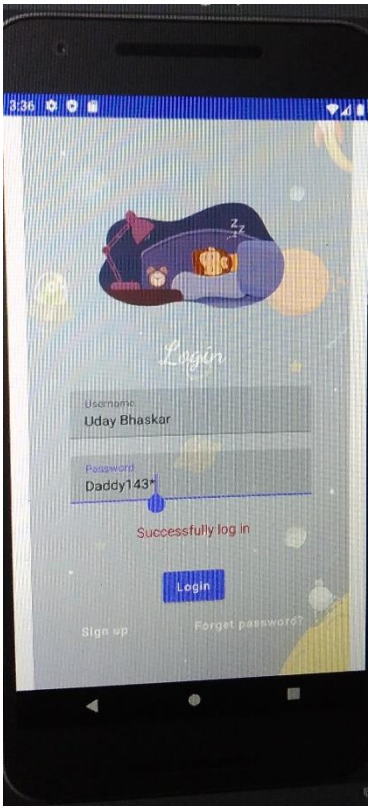
3.RESULTS

LOGIN PAGE

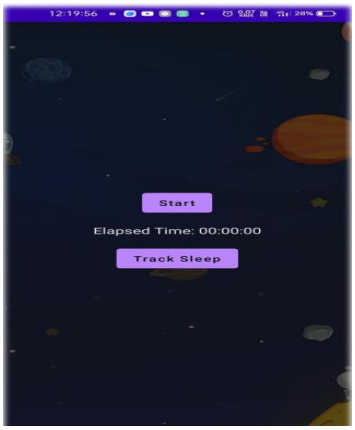


REGISTER PAGE:

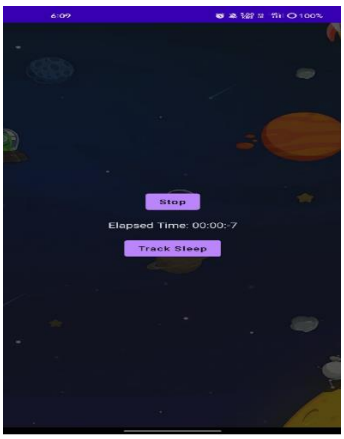




START TIMER PAGE:



STOP TIMER:



TRACK TIMING PAGE:

4.ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

1. Helps You Better Understand Your Sleep Patterns In order to improve your sleep patterns, you need to understand them. ..

2. Tracks Your Different Sleep Phases There are generally four sleep phases: awake, light, deep, and REM.

3. Captures Possible Sleep Interruptions and Disruptions .

4. Gently Wakes You Up at the Ideal Time ..

5. Convenient Compared to Wearing a Sleep Tracker Device ...

5.DISADVANTAGES:

1. Hinders Sleep More Than It Helps Sure, you can use asleep tracker app to track multiple aspects of your sleep, like your sleep phases, sleep debt, and sleep goals. ...

2. Promotes Nighttime Screen Use Before Bed ...

3. Incorrect or Unreliable Sleep Tracking Data .

4. Creates an Unhealthy Sleep Obsession ...

5. Unable to Track Heart Rate and Movements ..

6.APPLICATION:

Personal sleep monitoring: Users can track their own sleep patterns, including the duration, quality, and consistency of their sleep. By monitoring their sleep over time, users can identify factors that may be affecting theirsleep and make changes to improve their sleep quality.

Sleep disorder management: People who suffer from sleep disorders such as insomnia or sleep apnea can use the sleep tracking app to monitor their sleep patterns and identify potential triggers or factors that may be exacerbating their condition. They can share the data with their healthcare provider to help with diagnosis and treatment.

Athlete performance tracking: Athletes can use the app to track their sleep quality and duration to optimize their performance. Getting adequate rest is crucial for recovery and athletic performance, and the app can helpathletes identify areas where they need to improve their sleep habits.

Workplace productivity: Employers can encourage their employees to use the app to monitor their sleep and identify factors that may be affecting their productivity.

By promoting better sleep habits, employers can improvethe health and well-being of their workforce and increaseproductivity.

Sleep research: Sleep researchers can use the app to collect data on sleep patterns and behavior. The app can provide a large dataset of sleep information, which can be used for research and analysis to better understand the science of sleep and its impact on health and well-being.

Wellness and lifestyle tracking: People who are interestedin wellness and lifestyle improvement can use the app to monitor their sleep habits as part of a larger wellness program. By tracking their sleep, they can identify areas

where they need to improve and develop healthier habits.

Overall, a sleep tracking app for Android can be used in a variety of contexts, from personal health and wellness to scientific research and workplace productivity. The app has the potential to help users achieve a better night's

rest and improve their overall health and well-being.

CONCLUSION:

Based on current research, sleep tracking apps can be useful tools for monitoring and improving sleep habits.

These apps can provide information about sleep patterns,duration, and quality, as well as offer suggestions for improving sleep hygiene

However, it is important to note that sleep tracking apps may not always be accurate and can sometimes provide misleading information. Additionally, relying solely on technology to improve sleep habits may not be the most effective approach, as addressing underlying lifestyle factors such as diet, exercise, and stress management may also be necessary.

Overall, if used in conjunction with other lifestyle changes and under the guidance of a healthcare professional, sleep tracking apps can be a useful tool for promoting better sleep and overall health. When choosing a sleep trackingapp, it is important to consider factors such as accuracy, ease of use, and privacy policies.

7.FUTURE SCOPE:

The future scope of sleep tracking apps for Android is quite promising. As technology advances, sleep tracking apps are likely to become more accurate and sophisticated, potentially incorporating features such as wearable sensors, machine learning algorithms, and personalized recommendations based on individual sleeppatterns and habits.

Moreover, sleep tracking apps are likely to become more integrated with other health and wellness apps, allowing for a more holistic approach to improving overall health.

For example, sleep tracking data could be used to inform dietary recommendations or suggest personalized exercise routines

Furthermore, as the importance of sleep on overall health becomes more widely recognized, sleep tracking apps may become more mainstream and integrated

into healthcare systems, potentially leading to earlier detection and intervention for sleep-related disorders.

Overall, the future scope of sleep tracking apps is vast, and as technology continues to advance, these apps are likely to become even more valuable tools for promoting better sleep and overall health.

8.APPENDIX

BUILD GRADLE:

```
buildscript {
 ext {
 compose_ui_version = '1.2.0'
 }
}// Top-level build file where you can add
configuration options common to all subprojects/modules.
plugins {
 id 'com.android.application' version '7.4.2'
apply false
 id 'com.android.library' version '7.4.2' apply
false
 id 'org.jetbrains.kotlin.android' version '1.7.0'
apply false
}
plugins {
 id 'com.android.application'
 id 'org.jetbrains.kotlin.android'
}
android {
```

```
 namespace 'com.example.projectone'
 compileSdk 33
 defaultConfig {
 applicationId "com.example.projectone"
 minSdk 24
 targetSdk 33
 versionCode 1
 versionName "1.0"
 testInstrumentationRunner
"androidx.test.runner.AndroidJUnitRunner"
 vectorDrawables {
 useSupportLibrary true
 }
 }
 buildTypes {
 release {
 minifyEnabled false
 proguardFiles
getDefaultProguardFile('proguard-androidoptimize.txt'), 'proguard-rules.pro'
 }
 }
 compileOptions {
 sourceCompatibility
JavaVersion.VERSION_1_8
 targetCompatibility
JavaVersion.VERSION_1_8
 }
```

```
 kotlinOptions {
 jvmTarget = '1.8'
 }
 buildFeatures {
 compose true
 }
 composeOptions {
 kotlinCompilerExtensionVersion '1.2.0'
 }
 packagingOptions {
 resources {
 excludes += '/META-INF/{AL2.0,LGPL2.1}'
 }
 }
}
dependencies {
 implementation 'androidx.core:core-ktx:1.7.0'
 implementation 'androidx.lifecycle:lifecycleruntime-ktx:2.3.1'
 implementation 'androidx.activity:activitycompose:1.3.1'
 implementation
"androidx.compose.ui:ui:$compose_ui_version"
 implementation "androidx.compose.ui:uitooling-
preview:$compose_ui_version"
 implementation
'androidx.compose.material:material:1.2.0'
 implementation 'androidx.room:roomcommon:2.5.0'
 implementation 'androidx.room:room-ktx:2.5.0'
 testImplementation 'junit:junit:4.13.2'
```

androidTestImplementation
'androidx.test.ext:junit:1.1.5'
androidTestImplementation
'androidx.test.espresso:espresso-core:3.5.1'
androidTestImplementation
"androidx.compose.ui:ui-testjunit4:$compose_ui_version"
debugImplementation "androidx.compose.ui:uitooling:$compose_ui_version"
debugImplementation "androidx.compose.ui:uitest-
manifest:$compose_ui_version"
}
ANDRIODMANIFEST .XML
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk
/res/android"

xmlns:tools="http://schemas.android.com/tools"
>
<application
android:allowBackup="true"

android:dataExtractionRules="@xml/data_extract
ion_rules"

android:fullBackupContent="@xml/backup_rules"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:supportsRtl="true"

```
 android:theme="@style/Theme.ProjectOne"
 tools:targetApi="31">
 <activity
 android:name=".TrackActivity"
 android:exported="false"

android:label="@string/title_activity_track"

android:theme="@style/Theme.ProjectOne" />
 <activity
 android:name=".MainActivity"
 android:exported="false"
 android:label="@string/app_name"

android:theme="@style/Theme.ProjectOne" />
 <activity
 android:name=".MainActivity2"
 android:exported="false"
 android:label="RegisterActivity"

android:theme="@style/Theme.ProjectOne" />
 <activity
 android:name=".LoginActivity"
 android:exported="true"
 android:label="@string/app_name"

android:theme="@style/Theme.ProjectOne">
```

```
 <intent-filter>
 <action
android:name="android.intent.action.MAIN" />
 <category
android:name="android.intent.category.LAUNCHE
R" />
 </intent-filter>
 </activity>
 </application>
</manifest>
```

MAINACTIVITY.KT

```
package com.example.projectone
import android.content.Context
import android.content.Intent
import android.icu.text.SimpleDateFormat
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.material.Button
import
androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
```

```
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.alpha
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import androidx.core.content.ContextCompat
import
com.example.projectone.ui.theme.ProjectOneThe
me
import java.util.*
class MainActivity : ComponentActivity() {
 private lateinit var databaseHelper:
TimeLogDatabaseHelper
 override fun onCreate(savedInstanceState:
Bundle?) {
 super.onCreate(savedInstanceState)
 databaseHelper =
TimeLogDatabaseHelper(this)
 databaseHelper.deleteAllData()
 setContent {
 ProjectOneTheme {
 // A surface container using the
'background' color from the theme
 Surface(
 modifier = Modifier.fillMaxSize(),
 color =
MaterialTheme.colors.background
```

```
 ) {
 MyScreen(this,databaseHelper)
 }
 }
 }
 }
}
@Composable
fun MyScreen(context: Context, databaseHelper:
TimeLogDatabaseHelper) {
 var startTime by remember {
mutableStateOf(0L) }
 var elapsedTime by remember {
mutableStateOf(0L) }
 var isRunning by remember {
mutableStateOf(false) }
 val imageModifier = Modifier
 Image(
 painterResource(id =
R.drawable.sleeptracking),
 contentScale = ContentScale.FillHeight,
 contentDescription = ""
,
 modifier = imageModifier
 .alpha(0.3F),
 )
 Column(
 modifier = Modifier.fillMaxSize(),
 horizontalAlignment =
```

```
Alignment.CenterHorizontally,
 verticalArrangement = Arrangement.Center
 ) {
 if (!isRunning) {
 Button(onClick = {
 startTime = System.currentTimeMillis()
 isRunning = true
 }) {
 Text("Start")

//databaseHelper.addTimeLog(startTime)
 }
 } else {
 Button(onClick = {
 elapsedTime =
System.currentTimeMillis()
 isRunning = false
 }) {
 Text("Stop")

databaseHelper.addTimeLog(elapsedTime,startTi
me)
 }
 }
 Spacer(modifier = Modifier.height(16.dp))
 Text(text = "Elapsed Time:
${formatTime(elapsedTime - startTime)}")
 Spacer(modifier = Modifier.height(16.dp))
```

```
 Button(onClick = { context.startActivity(
 Intent(
 context,
 TrackActivity::class.java
 )
 ) }) {
 Text(text = "Track Sleep")
 }
 }
}
private fun startTrackActivity(context: Context) {
 val intent = Intent(context,
TrackActivity::class.java)
 ContextCompat.startActivity(context, intent,
null)
}
fun getCurrentDateTime(): String {
 val dateFormat = SimpleDateFormat("yyyy-MMdd HH:mm:ss",
Locale.getDefault())
 val currentTime = System.currentTimeMillis()
 return dateFormat.format(Date(currentTime))
}
fun formatTime(timeInMillis: Long): String {
 val hours = (timeInMillis / (1000 * 60 * 60)) %
24
 val minutes = (timeInMillis / (1000 * 60)) % 60
 val seconds = (timeInMillis / 1000) % 60
 return String.format("%02d:%02d:%02d",
hours, minutes, seconds)
}
```