**A SLEEP TRACKING APP FOR A BETTER NIGHT’S REST**

**Team Leader:**

* + Mythili M

NM ID: 3E2BA1D3ED3EA05B87549A2035A2C3E9

REG NO:710022243026

**Team Members:**

* + Jayasree S

NM ID: F0F5009B2324D6188544BEA2299B3649

REG NO:710022243017

* + Sowndarya A

NM ID: 5065E77C1CD358296D34B12A5B445D5A

REG NO:710022243028

* + Rekha G

NM ID: 962623D525AED8A9ED94641C4765492E

REG NO:710022243013

**ABSTRACT**

The "Sleep Tracker" app is a mobile application developed using Android Jetpack Compose, designed to help users monitor and improve their sleep quality. This app enables users to start a timer when they are about to fall asleep and stop it upon waking, with the timer running seamlessly in the background. After waking, users can rate the quality of their sleep through an intuitive rating system, providing insights into how well-rested they feel. The app then displays a visual analysis of the previous night's sleep, helping users identify patterns and assess overall sleep quality. Utilizing Jetpack Compose, the app's UI is modern, responsive, and easy to navigate, while the underlying view model and Repository layers ensure efficient data handling and storage. By encouraging mindful sleep habits, the "Sleep Tracker" app aims to promote better mental and physical well-being for its users.

**PROJECT OBJECTIVES**

* **Automatic Timer:** The app should accurately track sleep duration by automatically starting and stopping the timer when the user falls asleep and wakes up.
* **Sleep Quality Tracking:** Users should be able to rate and monitor their sleep quality over time, allowing them to identify trends and patterns.
* **Background Operation:** The timer needs to run continuously in the background to ensure uninterrupted sleep tracking, even when the phone is locked.
* **Insights and Improvement:** Helps users evaluate their sleep habits and make informed decisions for better sleep health.

**DESCRIPTION**

* The **Sleep Tracker** app is a mobile tool designed to help people get a better night's rest by giving them insights into their sleep patterns. Built with **Android Jetpack Compose**, the app provides an easy way to track sleep duration, rate sleep quality, and see an analysis of past sleep sessions, all in one place.
* Here’s how it works: when users are ready to go to bed, they start a timer that keeps running until they wake up. This feature runs quietly in the background so users don’t have to worry about interruptions. In the morning, they can rate how well they slept, which helps the app build a profile of their sleep quality over time. By analyzing this data, the **Sleep Tracker** app provides personalized insights, showing trends in sleep duration and quality that users can use to identify any patterns or habits affecting their rest.
* The app is designed to be simple and intuitive, making it easy to set up and use. With **Jetpack Compose** powering the interface, it’s modern, responsive, and looks great on any screen. Behind the scenes, features like ViewModel and Repository layers keep the app running smoothly and handling data reliably, even in the background. For anyone looking to better understand their sleep and develop healthier habits, the **Sleep Tracker** app offers a practical and accessible solution.

**SOLUTION**

* **Background Timer Accuracy**: Used **WorkManager** and **Foreground Services** to keep the timer running in the background.
* **Learning Jetpack Compose**: Followed tutorials and documentation to master Jetpack Compose for dynamic UIs.
* **Efficient Data Storage**: Used **Room Database** and **coroutines** for fast, asynchronous data handling.
* **State Management**: Managed app states with **ViewModel** and **LiveData** for consistency across screens.
* **Permissions and Background Limitations**: Requested permissions and guided users to prevent battery optimizations.
* **Data Visualization**: Used **Canvas API** in Jetpack Compose for clear and simple sleep data visuals.

**KEY FUNCTIONALITIES**

* **Sleep Tracking:** Users can start a sleep timer when they go to bed and stop it when they wakeup. The timer runs in the background, ensuring uninterrupted tracking.
* **Sleep Quality Rating:** After stopping the timer, users can rate their sleep quality based on their experience.
* **Sleep Analysis:** The app provides an analysis of the user’s sleep patterns from the previous night, offering insights into sleep duration and quality.
* **User Interface:** Built using Android Jetpack Compose, the app features an intuitive and responsive design that enhances user experience.

**CODE (MAIN ACTIVITY)**

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.ProjectOneTheme

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 {

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,startTime)

}

}

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-MM-dd HH:mm:ss", Locale.getDefault())

val currentTime = System.currentTimeMillis()

return dateFormat.format(Date(currentTime))

}

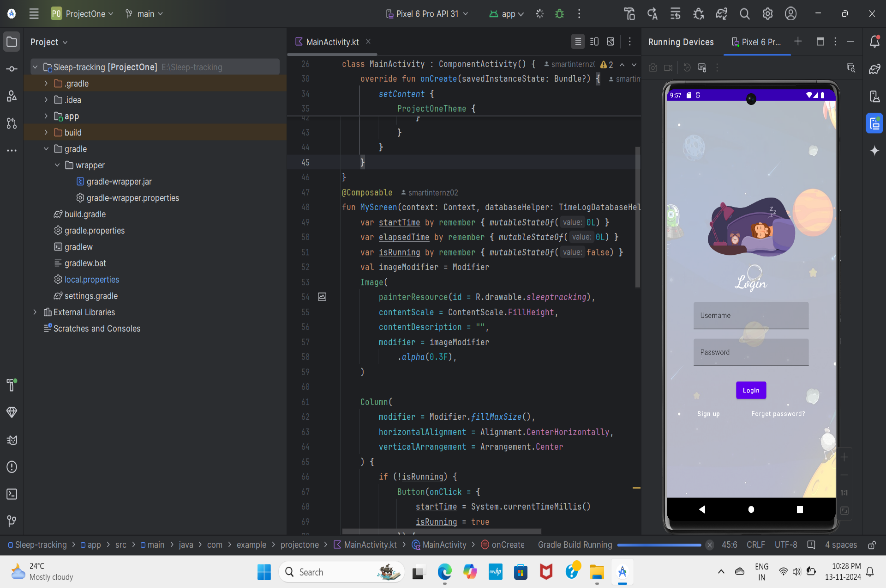
fun formatTime(timeInMillis: Long): String {

val hours = (timeInMillis / (1000 \* 60 \* 60)) % 24

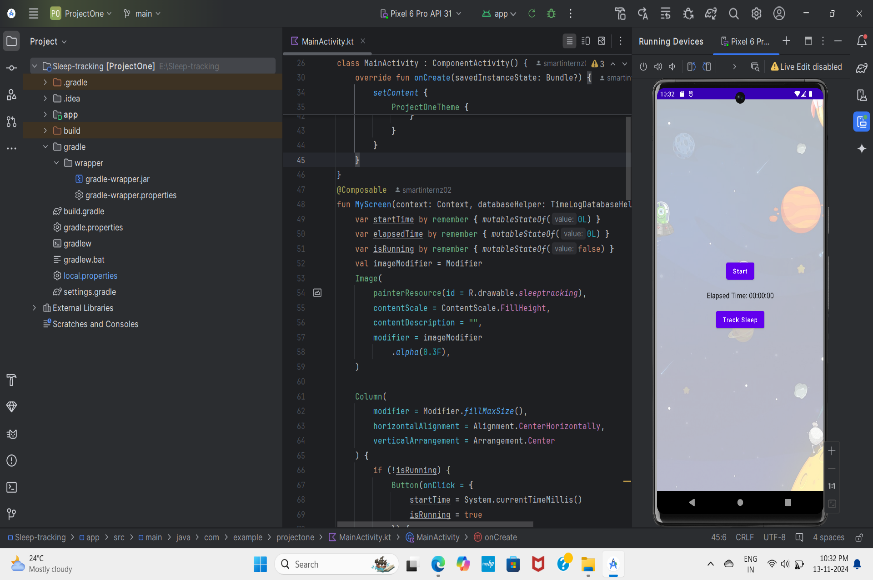
val minutes = (timeInMillis / (1000 \* 60)) % 60

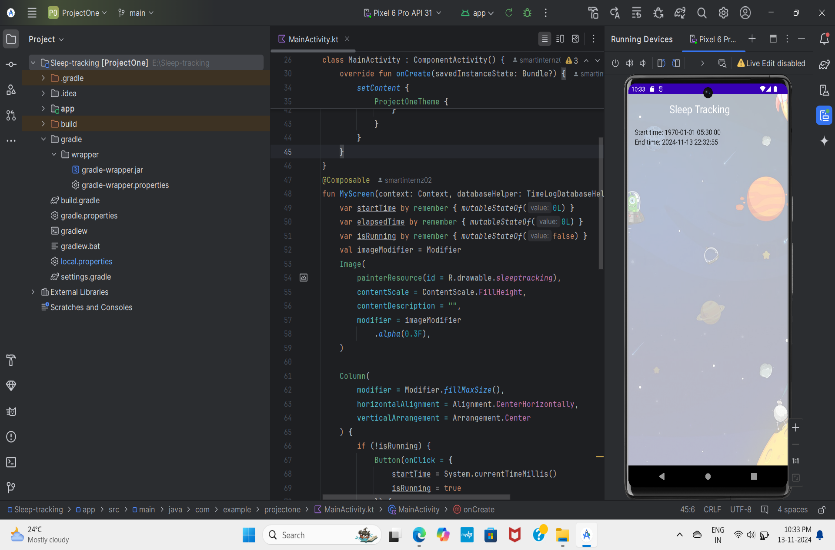
val seconds = (timeInMillis / 1000) % 60

return String.format("%02d:%02d:%02d", hours, minutes, seconds)

**OUTPUT**

Login page

Start sleep tracking



End sleep tracking

**REFERENCES:**

**Demo video link:**

<https://drive.google.com/file/d/1Xr-3dDiPT0R57rTxLGiEhNLPz-I9HtX5/view?usp=drive_link>

1. **Android Documentation for Jetpack Compose**:

* Official documentation explaining Jetpack Compose's features and usage:  
  <https://developer.android.com/jetpack/compose>

2. **WorkManager for Background Tasks**:

* Official Android documentation on WorkManager for handling background tasks:  
  <https://developer.android.com/topic/libraries/architecture/workmanager>

3. **Room Database for Data Storage**:

* Official guide on using Room Database for local data storage in Android apps:  
  <https://developer.android.com/training/data-storage/room>

4.**Android Sleep Tracking and Sleep Science**:

* A resource on the importance of sleep and how sleep tracking can help: <https://www.sleepfoundation.org/>
* A study on sleep quality and tracking apps:  
  <https://pubmed.ncbi.nlm.nih.gov/32520907/>

5.**Android Permissions and Background Tasks**:

* Guide on managing permissions in Android, especially for background processes:

<https://developer.android.com/guide/topics/permissions/overviw>

**CONCLUSION:**

The "Sleep Tracker" app uses Jetpack Compose and WorkManager to provide a seamless and reliable experience for tracking sleep. By addressing background task management, accurate data storage with Room Database, and real-time updates through View Model and LiveData, the app ensures precise monitoring of sleep duration and quality over time. With a clean interface and robust functionality, it highlights the importance of sleep for health, enabling users to make informed choices to improve their sleep habits. Despite Android’s background limitations and new UI frameworks, this project demonstrates the potential of modern Android tools for building user-friendly, powerful applications.