INTRODUCTION

1.1 OVERRVIEW

The Sleep Tracker project is an Android Kotlin-based application designed to help users track their sleep patterns. The app includes a user login and registration form for security purposes. Once the user logs in with their username and password, they will be directed to the main page, which contains two buttons: Start and Stop.

The Start button is used to initiate the tracking of the user's sleep when they are ready to go to bed. The Stop button is used to stop tracking once the user wakes up in the morning. The app will keep track of the minutes slept and store that data for the user to review later.

This Sleep Tracker application is an excellent tool for anyone interested in monitoring their sleep patterns and getting insights into their sleep quality. It provides a simple and easy-to-use interface that allows users to track their sleep with just a few taps on their mobile devices.

Purpose:

1.Monitor sleep patterns: The primary purpose of a sleep tracker is to monitor an individual's sleep patterns. By tracking the duration and quality of sleep, the app can provide valuable insights into how well a person is sleeping.

2.Identify sleep trends: A sleep tracker can help identify trends in a person's sleep patterns, such as how long it takes to fall asleep, how many times they wake up during the night, and how long they stay asleep.

3.Improve sleep quality: With the data provided by a sleep tracker, individuals can make lifestyle changes to improve their sleep quality. For example, if the data shows that a person has trouble falling asleep, they can adjust their bedtime routine to include relaxation techniques or avoid using electronic devices before bed.

4.Enhance overall health: Poor sleep quality can lead to a range of health issues, including weight gain, a weakened immune system, and an increased risk of chronic diseases such as diabetes and heart disease. By improving sleep quality, individuals can enhance their overall health and well-being.

5.Increase productivity: Good quality sleep is essential for productivity. A sleep tracker can help individuals identify areas where they can improve their sleep quality, which can result in improved productivity and focus throughout the day.

6.Track progress: A sleep tracker can help individuals track their progress over time. By monitoring changes in sleep patterns and habits, individuals can see the impact of their efforts to improve their sleep quality.

2.1 EMPATHY MAP:

Says:

"I have trouble sleeping through the night"

"I'm not sure how long I'm sleeping each night"

"I feel tired during the day"

Thinks:

"I need to get a better night's sleep to feel refreshed in the morning"

"I wonder if there's anything I can do to improve my sleep quality"

"I'm worried about the long-term effects of poor sleep on my health"

USER

Does:

Checks the time on their phone when they wake up during the night

Tries different sleeping positions to get comfortable

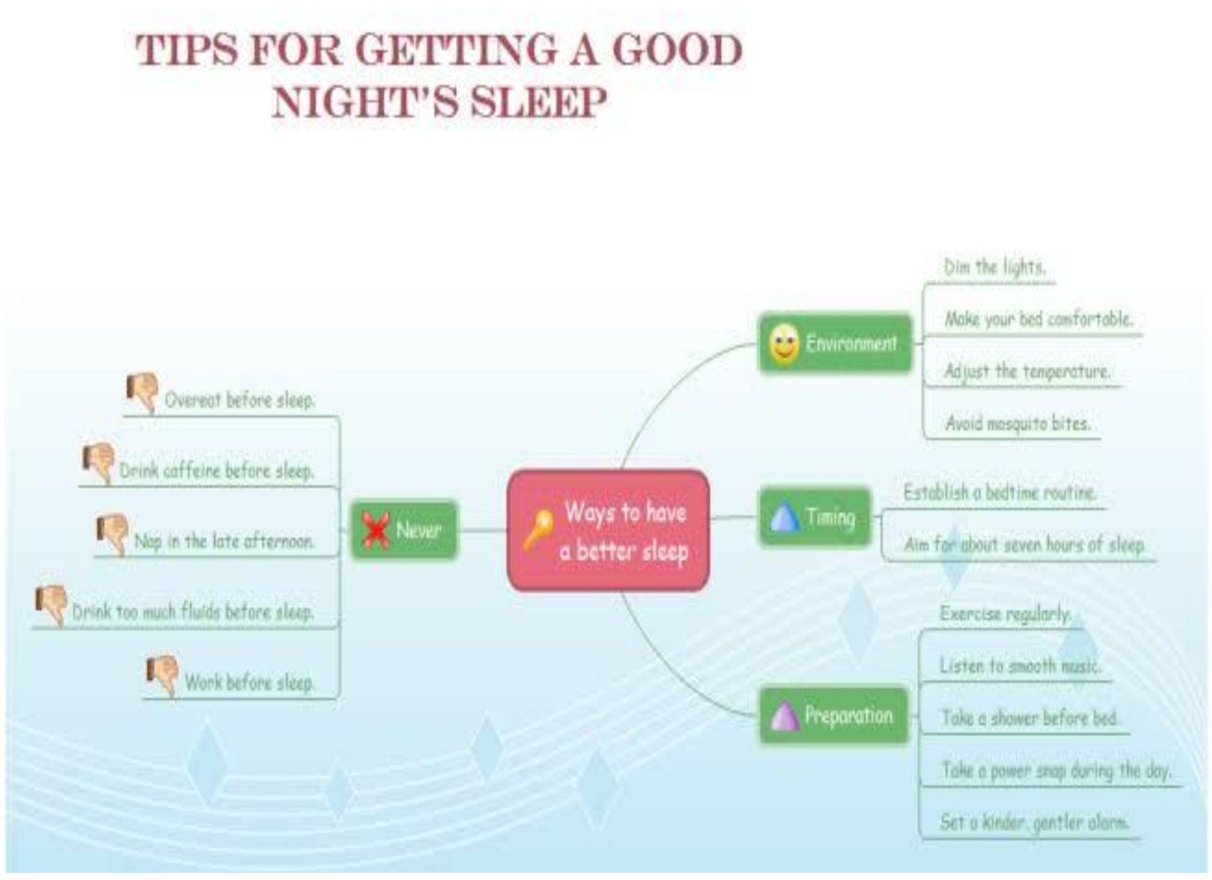
Uses relaxation techniques like deep breathing or meditation to fall asleep

Feels:

Frustrated when they can't fall asleep or stay asleep

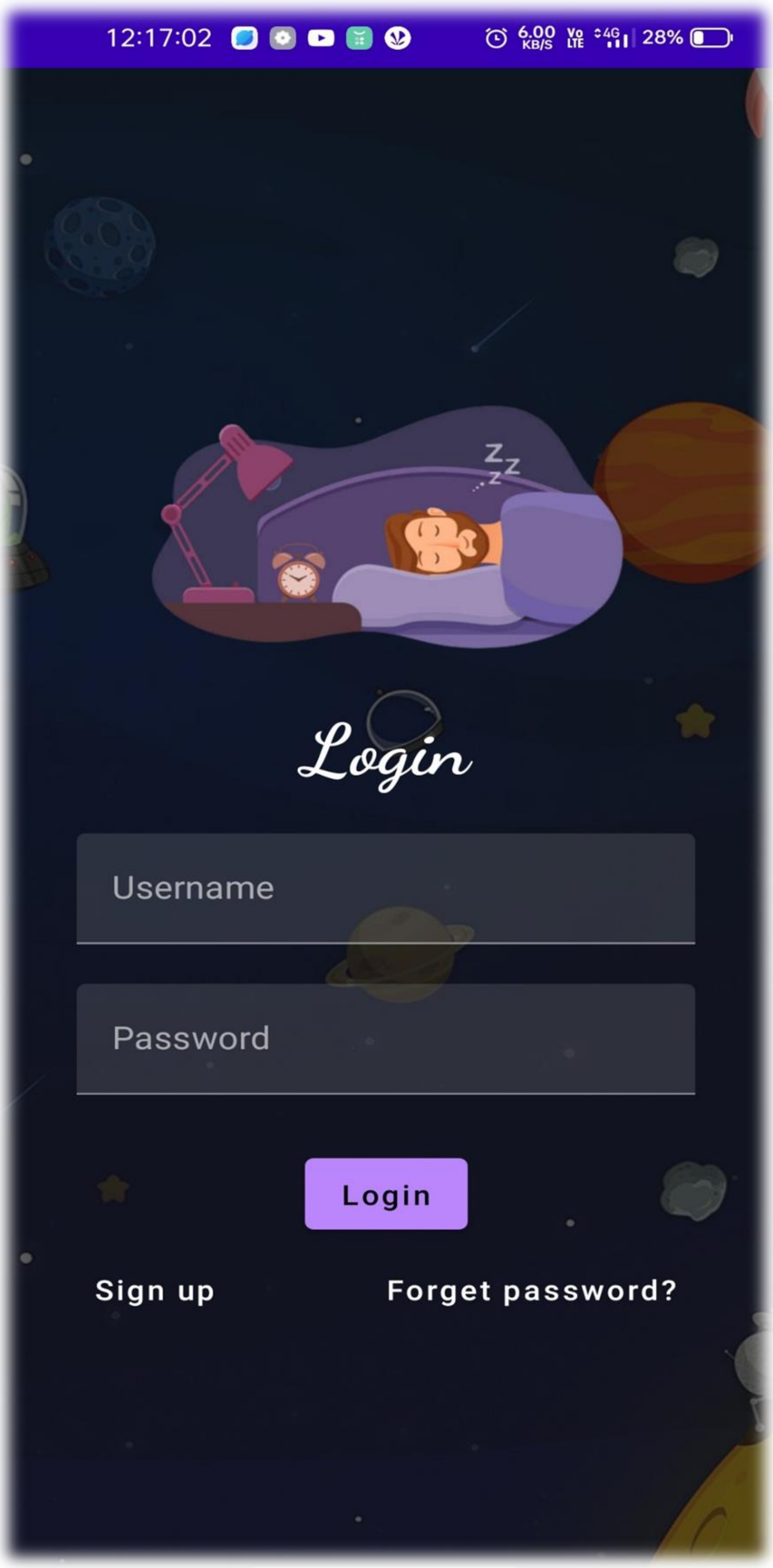
Anxious about not getting enough sleep

2.2 IDEATION & BRAINSTORMING MAP

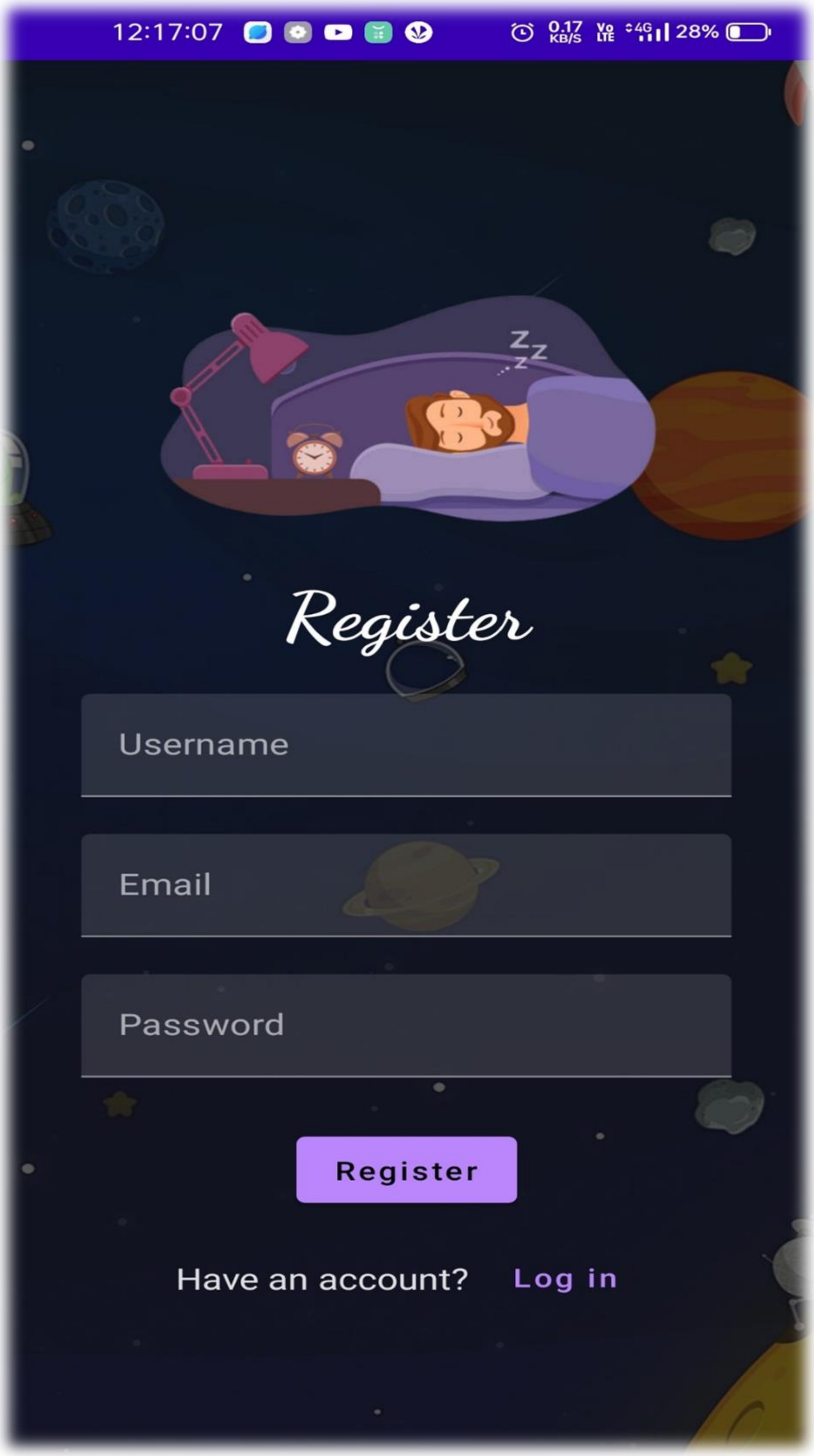

TIPS FOR GETTING A GOOD NIGHT'S SLEEP
Ways to have a better sleep
Never
Overeat before sleep.
Drink caffeine before sleep.
Nap in the late afternoon.
Drink too much fluids before sleep.
Work before sleep.
Environment
Dim the lights.
Make your bed comfortable.
Adjust the temperature.
Avoid mosquito bites.
Timing
Establish a bedtime routine.
Aim for about seven hours of sleep.
Preparation
Exercise regularly.
Listen to smooth music.
Take a shower before bed.
Take a power snap during the day.
Set a kinder, gentler alarm.

RESULT

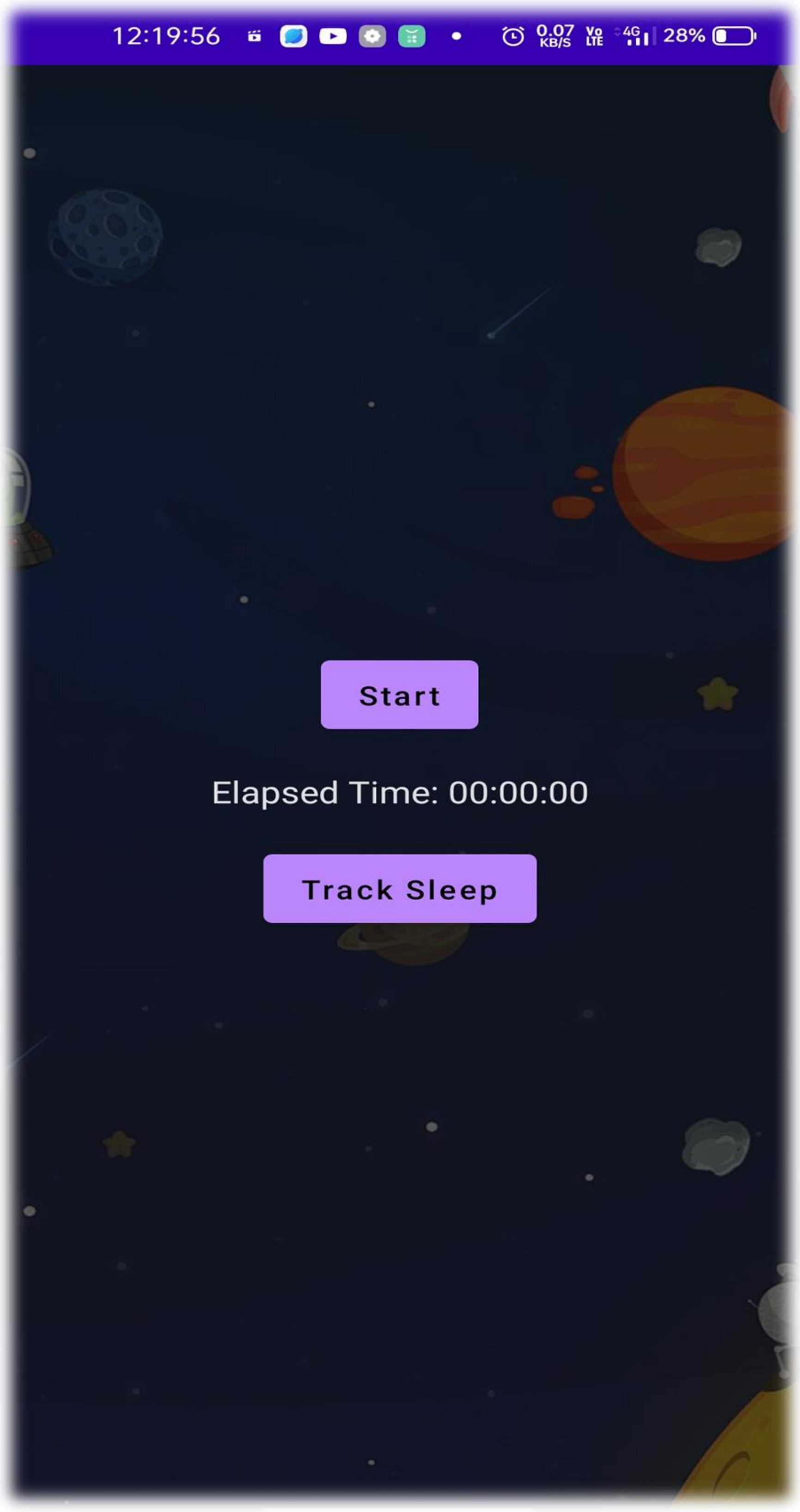
3.1 LOGIN PAGE



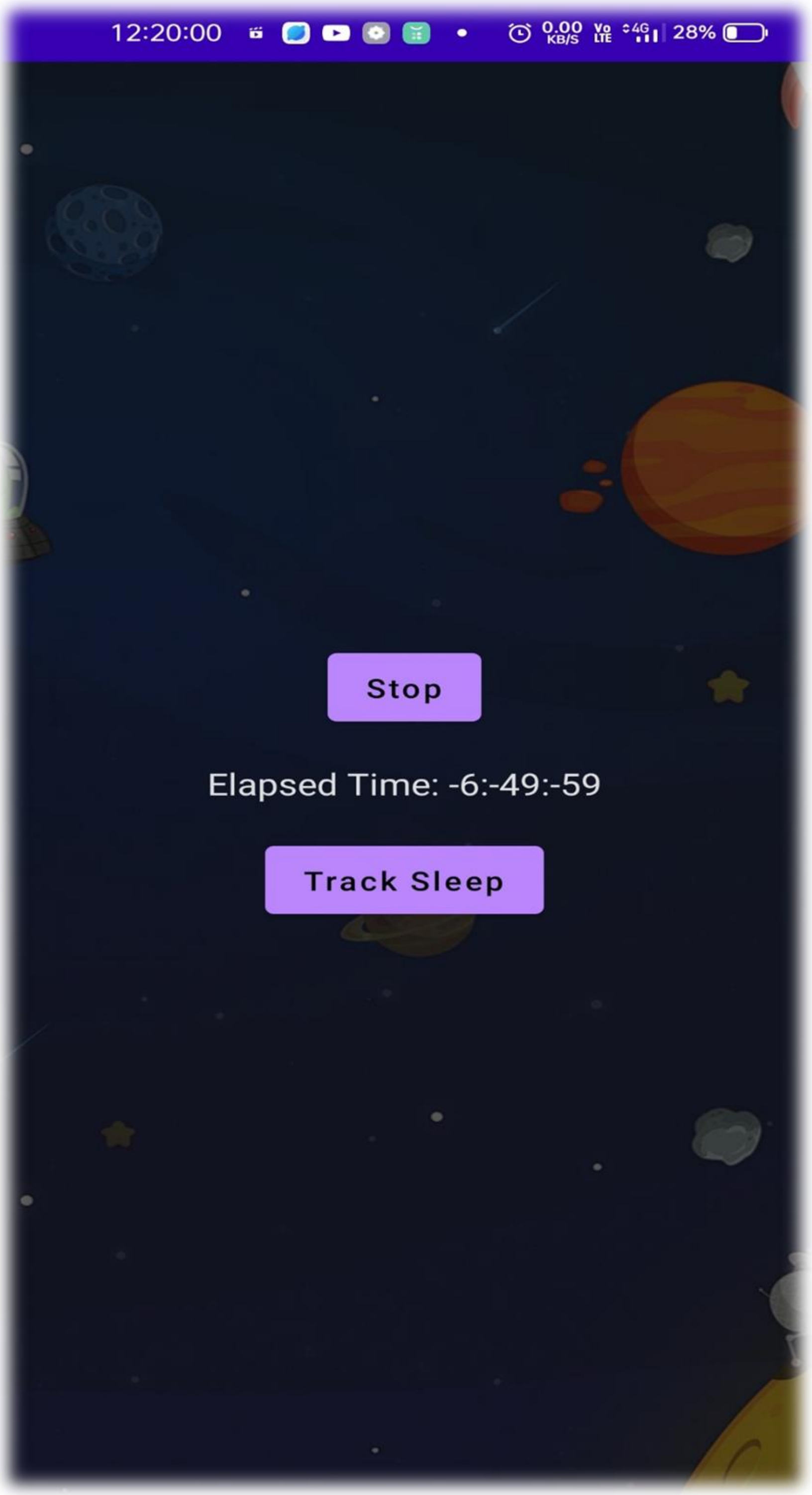
3.2 REGISTER PAGE:



3.3 START TIMER PAGE:

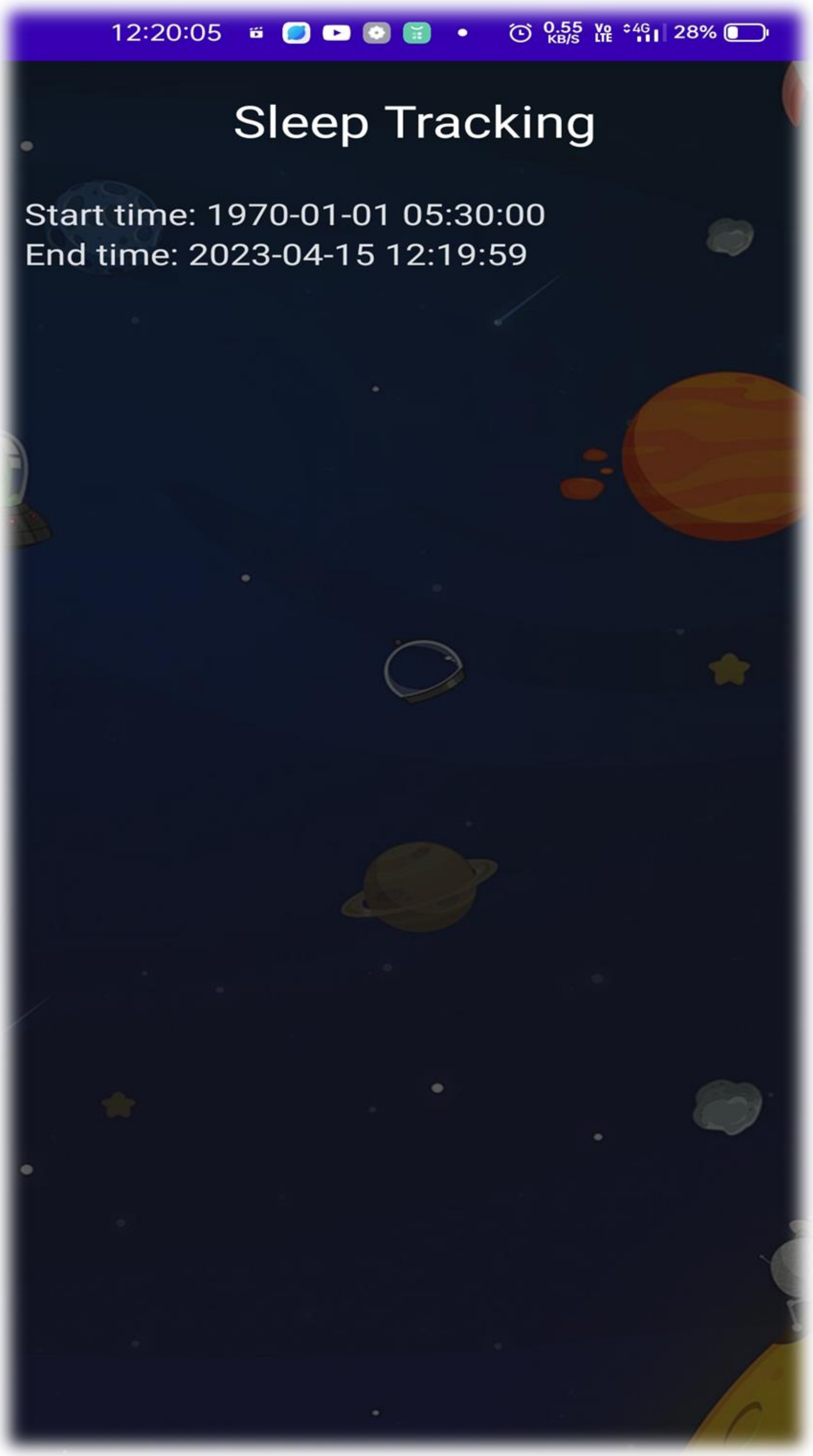


3.4 STOP TIMER:



Elapsed Time: -6:-49:-59

3.5 TRACK TIMING PAGE:


Sleep Tracking

Start time: 1970-01-01 05:30:00
End time: 2023-04-15 12:19:59

ADVANTAGES AND DISADVANTAGES

4.1 ADVANTAGES:

1.Improved Sleep Quality: Sleep tracker apps can help users identify patterns in their sleep and suggest changes to improve their quality of sleep. Users can also use the app to set goals and track their progress towards getting better sleep.

2.Better Understanding of Sleep Habits: Sleep tracker apps can provide users with a comprehensive view of their sleep habits, including how long it takes them to fall asleep, how often they wake up during the night, and how long they spend in each sleep stage. This information can help users understand their sleep patterns and make changes to improve them.

3.Personalized Insights: Many sleep tracker apps use artificial intelligence (AI) and machine learning algorithms to provide personalized insights into the user's sleep habits. These insights can help users understand what is affecting their sleep and how they can make changes to improve it.

4.Convenience: Sleep tracker apps are easy to use and can be accessed on smartphones and other devices. They can also be used in conjunction with other health and fitness apps to provide a more comprehensive view of the user's overall health.

4.2 DISADVANTAGES:

1.Reliability: Sleep tracker apps rely on sensors in the user's device to track their sleep. These sensors may not always accurately capture the user's movements, resulting in inaccurate data.

2.Battery Drain: Sleep tracker apps can be battery-intensive and may drain the user's device battery quickly. This can be inconvenient, especially if the user needs to use their device for other purposes during the day.

3.Inaccuracy: Sleep tracker apps may not accurately track all aspects of sleep, such as the user's breathing patterns, heart rate, or brain activity. This can result in incomplete or inaccurate data.

4.Overreliance on Technology: Some users may become overly reliant on sleep tracker apps to manage their sleep, which can lead to anxiety and stress if the app does not work as expected or if the user's sleep patterns change.

APPLICATION

5.1 APPLICATIONS:

Sleep trackers are applications that are designed to track and analyze your sleep patterns to provide insights into the quality and quantity of your sleep. Here are some common applications of sleep trackers:

Improve Sleep Quality: Sleep trackers can help you identify patterns in your sleep and suggest changes to improve your quality of sleep. By tracking how long you sleep, how often you wake up, and how much time you spend in each sleep stage, you can adjust your habits to improve your overall sleep quality.

Identify Sleep Disorders: Sleep trackers can help you identify if you have a sleep disorder such as insomnia or sleep apnea. The data collected by the app can be shared with your healthcare provider to help diagnose and treat these disorders.

Set Sleep Goals: Sleep trackers can help you set goals for how long you want to sleep and how many times you wake up during the night. You can track your progress toward these goals and adjust your habits to meet them.

Improve Overall Health: Sleep is an important factor in overall health and well-being. By tracking your sleep patterns, you can identify areas where you need to improve your habits and make changes that can lead to better health outcomes.

Track the Effectiveness of Interventions: If you're trying to improve your sleep through lifestyle changes or medical interventions, sleep trackers can help you track the effectiveness of these interventions. You can see how your sleep patterns change over time and adjust your interventions accordingly.

Overall, sleep trackers can be useful tools for improving sleep quality and identifying potential sleep disorders. However, it's important to remember that sleep trackers have limitations and should be used in conjunction with other methods for improving sleep and overall health.

CONCLUSION

6.1 CONCLUSION

In conclusion, sleep tracker apps for Android can be valuable tools for improving sleep quality and identifying potential sleep disorders. They offer a range of features including tracking sleep duration, sleep stages, and sleep quality, as well as providing personalized insights into sleep habits. By using sleep tracker apps, Android users can gain a better understanding of their sleep patterns and make changes to improve the quality of their sleep. However, it's important to be aware of the limitations of sleep trackers, such as potential inaccuracies and overreliance on technology. Overall, sleep tracker apps can be a helpful addition to a broader approach to improving sleep, including good sleep hygiene habits and seeking professional medical advice if necessary.

FUTURE SCOPE

6.1 FUTURE SCOPE:

The future scope of sleep tracking apps is vast and exciting. Here are some potential areas of development:

Integration with other health tracking apps: Sleep tracking apps may integrate with other health tracking apps, such as fitness or nutrition apps, to provide a more comprehensive view of overall health.

Real-time feedback: Sleep tracking apps may incorporate real-time feedback to help users adjust their sleep habits in real-time. This could include suggestions for adjusting room temperature, lighting, or sound levels to improve sleep quality.

Advanced sensors: Sleep tracking apps may incorporate advanced sensors, such as heart rate monitors, to provide more accurate data and insights into sleep patterns.

Integration with smart home technology: Sleep tracking apps may integrate with smart home technology, such as smart lights or thermostats, to create a more personalized sleep environment.

Personalized coaching: Sleep tracking apps may incorporate personalized coaching and support to help users make lasting changes to their sleep habits.

Better integration with wearable devices: Sleep tracking apps may better integrate with wearable devices, such as smartwatches or fitness trackers, to provide more accurate and comprehensive sleep data.

Overall, the future of sleep tracking apps is promising, with the potential for more personalized, accurate, and effective sleep tracking and coaching. As technology advances, sleep tracking apps may become an even more valuable tool for improving overall health and well-being.

APPENDIX

BUILD GRADLE:

```
buildscript {
    ext {
        compose_ui_version = '1.2.0'
    }
}// Top-level build file where you can add
configuration options common to all sub-
projects/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-android-
optimize.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:lifecycle-
runtime-ktx:2.3.1'
    implementation 'androidx.activity:activity-
compose:1.3.1'
    implementation
"androidx.compose.ui:ui:$compose_ui_version"
    implementation "androidx.compose.ui:ui-
tooling-preview:$compose_ui_version"
    implementation
'androidx.compose.material:material:1.2.0'
    implementation 'androidx.room:room-
common: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-test-
junit4:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-
tooling:$compose_ui_version"
```

```
    debugImplementation "androidx.compose.ui:ui-
test-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-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)
}
```