

SLEEP TRACKING APP FOR A BETTER NIGHT'S REST

1.PROJECT DESCRIPTION

Many individuals struggle to get consistent, high-quality sleep due to various factors, such as stress, irregular schedules, and lack of awareness about their sleep patterns. Although there is widespread awareness of the importance of sleep, people rarely have tools to track their sleep stages or pinpoint disruptions and habits that contribute to poor rest. Current solutions often lack personalization, actionable insights, or seamless integration with users' lifestyles. Thus, there is a need for a comprehensive yet user-friendly sleep tracking application that can accurately monitor sleep patterns, provide data-driven insights, and suggest practical recommendations for enhancing sleep quality.

Sleep is one of the most critical factors for physical health, cognitive function, and emotional stability. However, many people struggle to maintain consistent, high-quality sleep due to stress, irregular schedules, and lifestyle habits that disrupt rest. While there is widespread awareness of the importance of sleep, few people have tools that can accurately track their sleep patterns and provide personalized insights into factors affecting their sleep quality. Most current solutions provide basic sleep data but lack personalization, actionable insights, or seamless integration with daily life, leaving users unsure of how to make meaningful improvements. Dreamy Tym is an Android-based sleep tracking app designed to help users achieve restful, high-quality sleep each night. Built with simplicity and user-friendliness in mind, Dreamy Tym provides users with the essential tools to monitor and enhance their sleep patterns, aiming to improve overall wellness through better night rest. By offering a clear and accessible interface, Dreamy Tym makes it easy for users to establish a healthy sleep routine. Recognizing the impact of sleep on physical and mental health, Dreamy Tym focuses on delivering a streamlined experience for tracking sleep habits and patterns. Unlike complex sleep apps, Dreamy Tym keeps its interface intuitive and easy to navigate, featuring a straightforward registration, login and tracking page. Users can securely create an account, log in, and track their sleep data to gain insights into their nightly rest.

1.1 KEY FUNCTIONALITIES

- **Sleep Cycle Analysis:** Monitors sleep stages, including light, deep, and REM sleep.
- **Sleep Disruption Detection:** Identifies disturbances, such as restlessness or waking episodes.
- **Personalized Recommendations:** Offers insights based on individual sleep data and trends.
- **Environmental Tracking:** Assesses room conditions, like light and noise, using device sensors.
- **Goal Setting and Progress Tracking:** Encourages users to set sleep goals and track improvements over time.

2. OBJECTIVES

The main objectives of the Sleep Tracking App project are as follows:

- **Accurately Track Sleep Stages:** Use device sensors to measure and analyze sleep stages and detect disturbances.
- **Provide Personalized Insights and Recommendations:** Offer actionable insights to improve sleep quality based on individual patterns.
- **Encourage Healthy Sleep Habits:** Allow users to set goals and track their progress over time.
- **Ensure Ease of Use:** Develop a user-friendly interface that requires minimal setup and interaction.

3. METHODOLOGY

The development of the Sleep Tracking App, Dreamy Tym, will follow an agile methodology, prioritizing flexibility and iterative development to enhance features and incorporate user feedback efficiently. This approach allows the Android development team to adapt quickly, ensuring the app meets evolving user needs and remains competitive.

Step-by-Step Approach

3.1 Requirements Gathering

- **User Research:** Conduct surveys, interviews, and market analysis to understand user preferences, pain points, and needs in sleep tracking.
- **Competitor Analysis:** Study leading sleep tracking apps to identify essential features, strengths, and gaps, defining a unique value proposition for Dreamy Tym.
- **Feature Finalization:** Compile insights to establish key functionalities and features, focusing on ease of use, user engagement, and core sleep tracking capabilities.

3.2 Design Phase

- **UI/UX Design:**

- Use Adobe XD or Figma to create a simple, intuitive interface that provides users with a clear view of their sleep metrics.
 - Design the following components:

- ▢ **Dashboard:** Displays essential sleep data like total sleep time, sleep stages (light, deep, REM), and a sleep quality score.
 - ▢ **Progress Reports:** Visualizes sleep trends and patterns over time through graphs, allowing users to track improvements.
 - ▢ **Personalized Recommendations:** Offers tailored suggestions for sleep improvement based on user habits and sleep data.

- **System Architecture:**

- Define the app's structure for a smooth user experience, data flow, and integration with Android components.
 - Architecture details include:

- ▢ **API Integration:** Define API calls for wearable device integration (like Google Fit) to gather real-time sleep data.
 - ▢ **Data Management:** Plan data handling across local storage and cloud backup, ensuring data consistency and accessibility.
 - ▢ **User Data Security:** Implement encryption and adhere to best practices in Android security to protect user privacy.

3.3 Development Phase

- **Frontend Development in Android Studio:**

- Use Android Studio and Kotlin for frontend development, focusing on Material Design for a visually appealing and responsive interface.
 - Implement XML layouts for core UI elements like registration, login, and sleep tracking screens.
 - Integrate RecyclerView for dynamic data display, such as sleep history and progress tracking.

- **Backend Development:**

- Set up Firebase as the backend to handle user authentication, data storage, and real-time data synchronization.
 - Use Firebase Realtime Database or Firestore to securely store and retrieve user sleep data and metrics.

- **Sleep Pattern Algorithms:**

- Develop algorithms to analyze sleep stages based on sensor data (e.g., motion, heart rate), detecting disruptions and generating insights.
- Use Google ML Kit or TensorFlow Lite for Android to build machine learning models that improve accuracy in sleep stage detection over time.

3.4 Testing Phase

- **Unit Testing:**
 - Conduct unit tests within Android Studio using JUnit to validate individual components, such as data capture accuracy and user interaction responses.
 - Test algorithms to ensure accurate detection and classification of sleep patterns and disruptions.
- **Integration Testing:**
 - Perform integration tests to ensure smooth interactions between frontend and backend modules, focusing on data synchronization and API performance.
 - Test integration with Google Fit or other compatible devices for seamless data flow.
- **User Testing:**
 - Gather a test group of users to provide feedback on app usability, interface design, and sleep analysis accuracy.
 - Use insights to refine app features, addressing any interface issues and enhancing the accuracy of sleep data interpretation.

3.5 Deployment

- **App Launch:**
 - Prepare an APK for release on the Google Play Store, optimizing the app for various Android devices and OS versions.
 - Set up Google Play Console to manage app listing, track user engagement, and monitor download statistics.
- **User Feedback Monitoring:**
 - Monitor user reviews, feedback, and crash reports in the Google Play Console to gather insights for post-launch improvements.
 - Collect and analyze usage data to identify opportunities for feature enhancements and performance improvements.

3.6 Maintenance and Updates

- **Feature Expansion:** Regularly add new functionalities, such as advanced sleep metrics or integration with new wearable devices, to enhance the app's value over time.
- **Bug Fixes and Performance Optimization:** Continuously address bugs and optimize performance based on user feedback, ensuring the app remains stable and user-friendly.

- **User Engagement and Support:** Build a support system within the app and encourage feedback to maintain high user engagement and satisfaction.

4.CODE IMPLEMENTATION

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.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 {
                // 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,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) }

```

AndroidManifest.xml

```

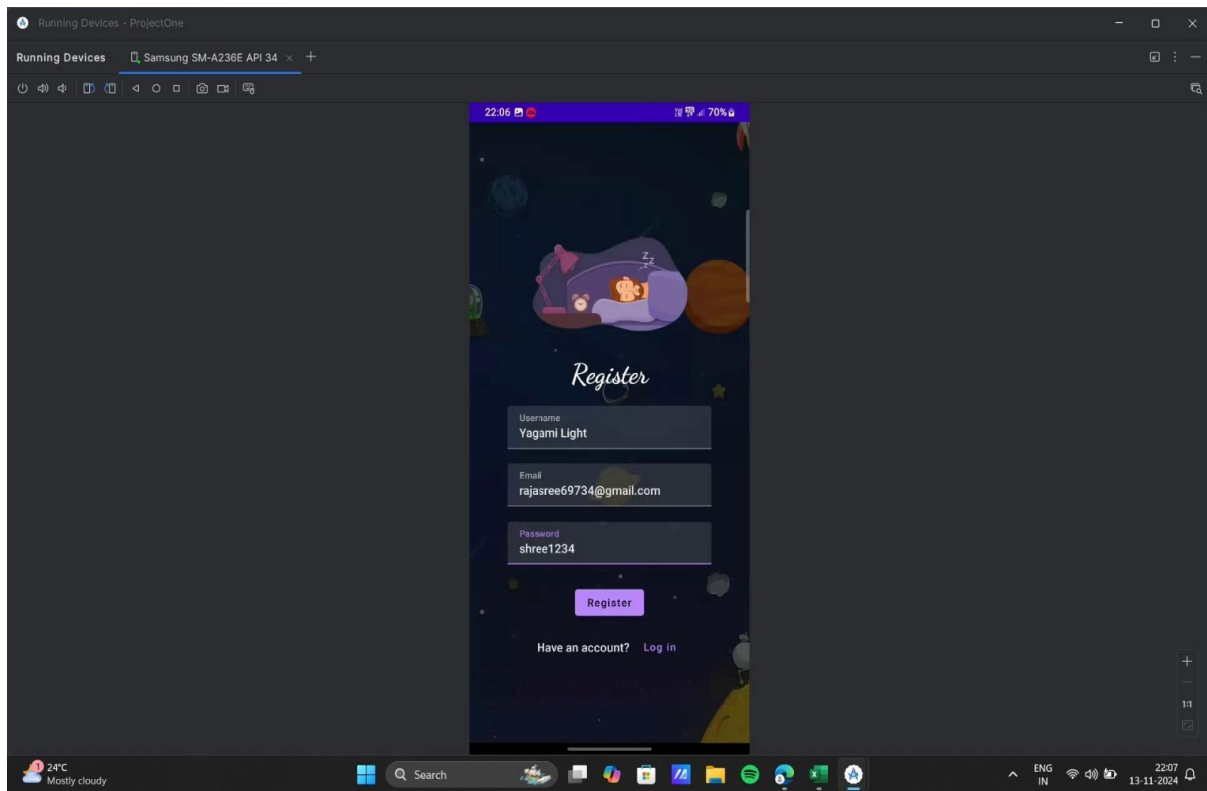
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"

```



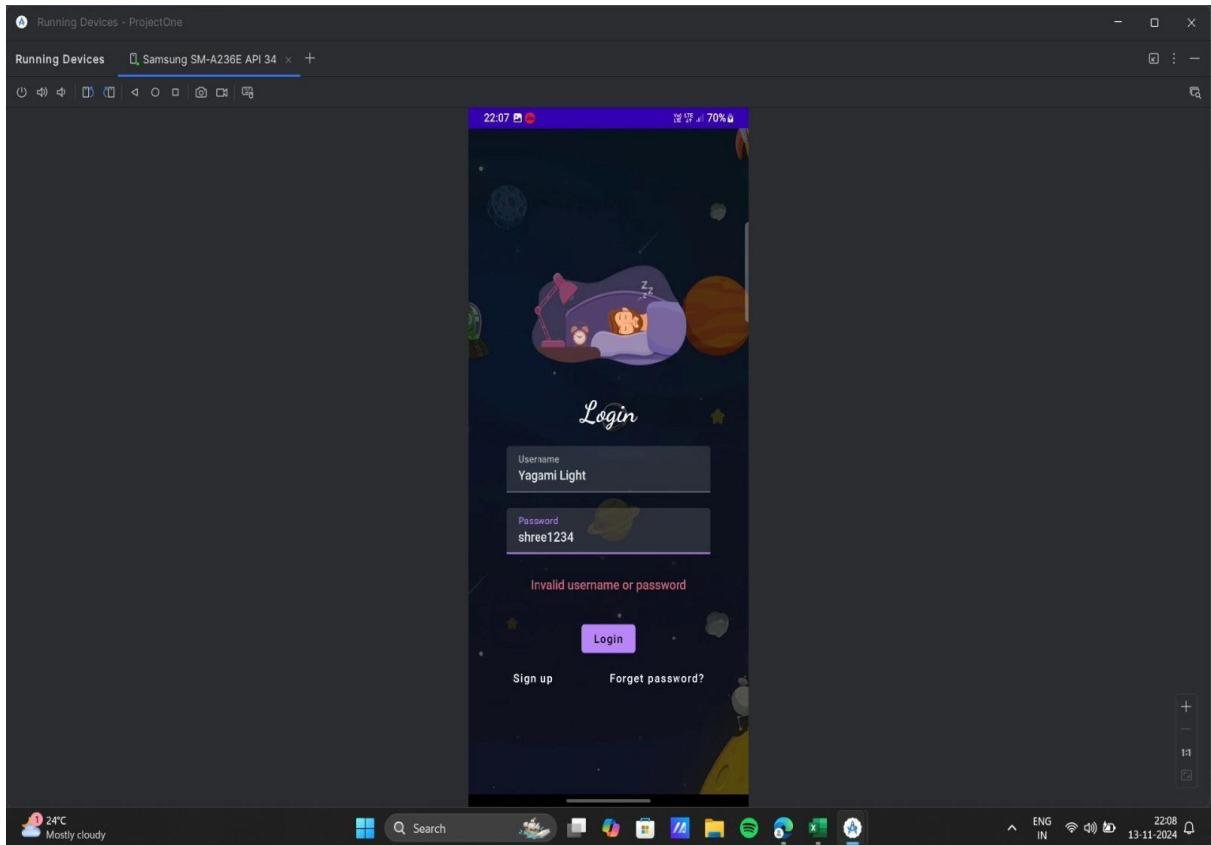
```
    android:theme="@style/Theme.ProjectOne"
    tools:targetApi="31">
        <activity        android:name=".TrackActivity"
        android:exported="false"
        android:label="@string/title_activity_track"
        android:theme="@style/Theme.ProjectOne" />
        <activity        android:name=".MainActivity"
        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" />
        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.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Output Registration page



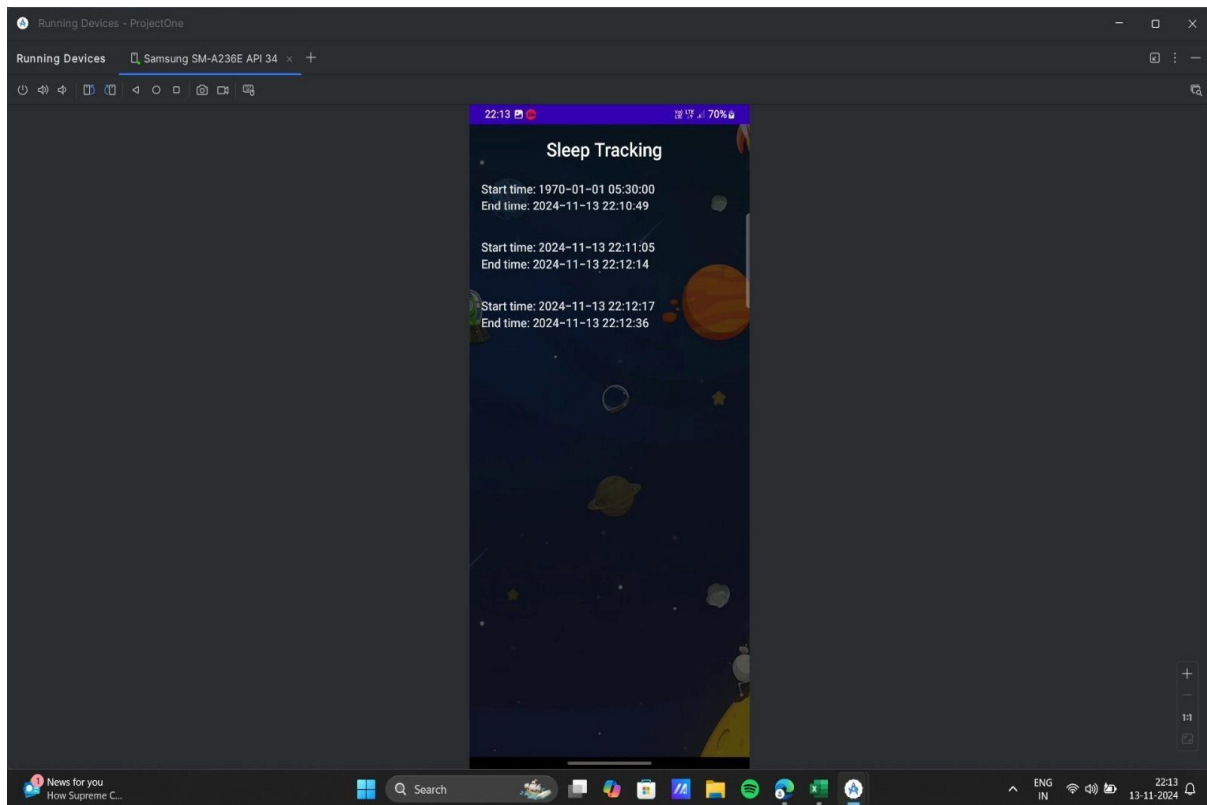
This registration page invites users to create an account in a sleep-tracking app designed to enhance restful nights. Users can enter a username, email, and password to get started. The background image adds a cozy, sleep-friendly aesthetic, aligning with the app's purpose of promoting better sleep and relaxation for users.

Login page



This login page allows registered users to access their accounts on the sleep-tracking app. Users enter their username and password, and they receive helpful feedback if credentials are incorrect. Options to "Sign up" or "Forget password?" provide smooth navigation for new users and those needing assistance, promoting an intuitive user experience.

Sleep-Tracking



This "Sleep tracking" screen allows users to monitor their sleep duration by pressing the "Start" button, which begins tracking their sleep time. The elapsed time is displayed on the screen, helping users keep track of their rest periods. When ready, users can press "Track Sleep" to save or analyze their data. This simple interface promotes ease of use, making it straightforward for users to track and improve their sleep habits effectively.

Demo video Link

https://drive.google.com/file/d/18UigRwvHN8tb8Cm68UZfHbvdJa_L-0Uu/view?usp=drive_link

Github Link <https://github.com/RAJASHREE10/Androidapp>

CONCLUSION

Dreamy Tym's capabilities extend beyond simple sleep tracking. The app analyzes sleep stages (light, deep, and REM) to offer a detailed view of the user's nightly rest, which can highlight specific disruptions or trends affecting sleep quality. This data, in turn, supports the app's personalized recommendation engine, which delivers actionable advice tailored to each user's unique sleep profile. For example, Dreamy Tym might suggest adjustments in bedtime routines, screen time, or caffeine intake based on individual sleep patterns.

With features like progress tracking, users can monitor their sleep quality over time, observing how adjustments in lifestyle or routine affect their rest. These visual trends and progress reports motivate users to maintain healthy habits, giving them a sense of control and progress as they work toward consistent, rejuvenating sleep. This continuous tracking and feedback loop fosters a proactive approach to health, as users gain insights into how even minor changes can lead to significant improvements in sleep and daily productivity.

Ultimately, Dreamy Tym's design emphasizes ease of use and accessibility, making it suitable for individuals at any level of familiarity with sleep tracking technology. Its focus on a userfriendly experience encourages people to regularly engage with the app, building long-term, healthy habits around sleep. In promoting better sleep hygiene and fostering awareness, Dreamy Tym offers a powerful tool for achieving restful, consistent, and restorative sleep—laying the foundation for improved physical and mental well-being.