Project Report

Team-591086

1. **INTRODUCTION**
2. **Project Overview**

Sleep tracking apps can be a helpful tool for improving your sleep quality. By tracking your sleep patterns over time, you can identify trends and patterns. This information can help you make changes to your sleep habits, such as going to bed and waking up at the same time each day, creating a relaxing bedtime routine, and avoiding caffeine and alcohol before bed.

3. **Project Overview**

Sleep tracking apps are designed to help you track your sleep patterns and identify areas where you can improve your sleep quality. They can help you:

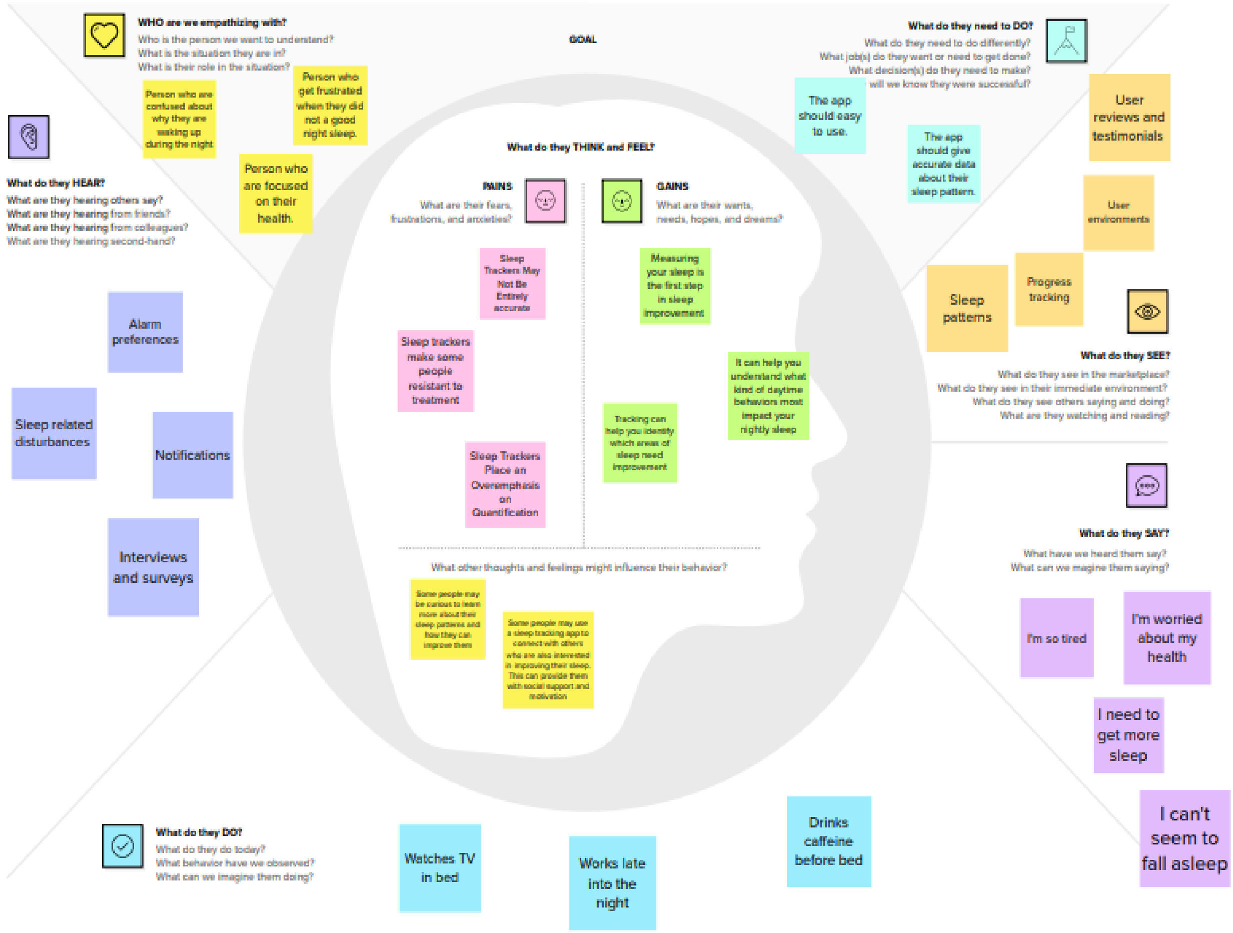
- Understand your sleep patterns
- Identify potential sleep problems
- Set and track sleep goals
- Improve your sleep habits

4. **LITERATURE SURVEY**
5. **Existing Problem**

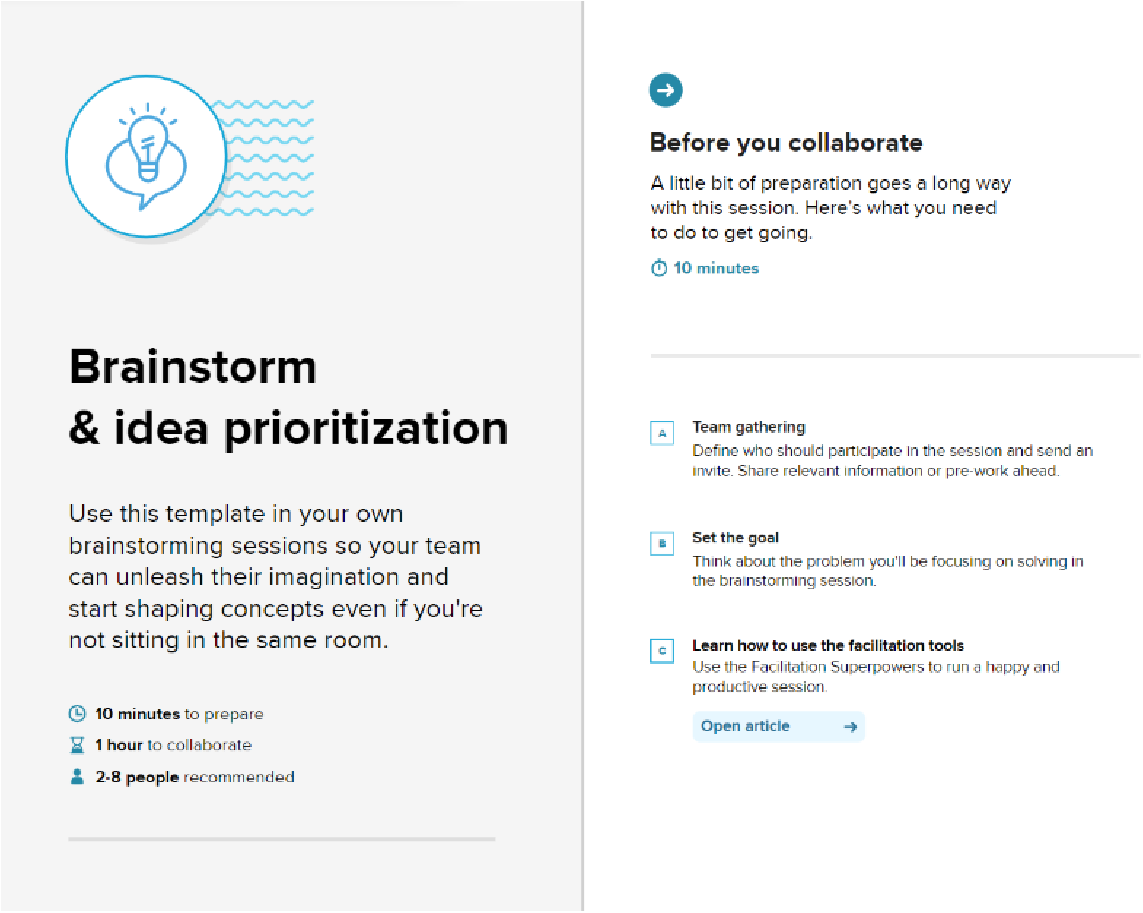
The existing problem with sleep tracking apps is that these apps rely on various methods like smartphone sensors, wearable watches, sound analysis, etc. to track sleep parameters. However, these methods can be susceptible to noise and interference, leading to inaccurate sleep data.

6. IDEATION & PROPOSED SOLUTION

3.1 Empathy map canvas


WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
Person who are confused about why they are waking up during the night
Person who get frustrated when they did not a good night sleep.
Person who are focused on their health.
GOAL
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
will we know they were successful?
The app should easy to use.
The app should give accurate data about their sleep pattern.
User reviews and testimonials
User environments
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
Alarm preferences
Sleep related disturbances
Notifications
Interviews and surveys
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
Sleep Trackers May Not Be Entirely accurate
Sleep trackers make some people resistant to treatment
Sleep Trackers Place an Overemphasis on Quantification
GAINS
What are their wants, needs, hopes, and dreams?
Measuring your sleep is the first step in sleep improvement
It can help you understand what kind of daytime behaviors most impact your nightly sleep
Tracking can help you identify which areas of sleep need improvement
What other thoughts and feelings might influence their behavior?
Some people may be curious to learn more about their sleep patterns and how they can improve them
Some people may use a sleep tracking app to connect with others who are also interested in improving their sleep. This can provide them with social support and motivation
Sleep patterns
Progress tracking
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
What do they SAY?
What have we heard them say?
What can we magine them saying?
I'm so tired
I'm worried about my health
I need to get more sleep
I can't seem to fall asleep
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?
Watches TV in bed
Works late into the night
Drinks caffeine before bed

3.2 Ideation & Brainstorm



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

⏱ **10 minutes**

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

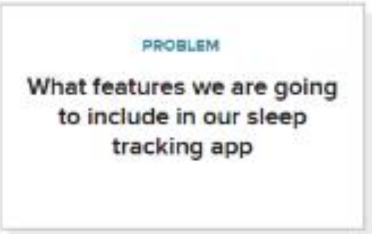
C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

1

Define your problem statement

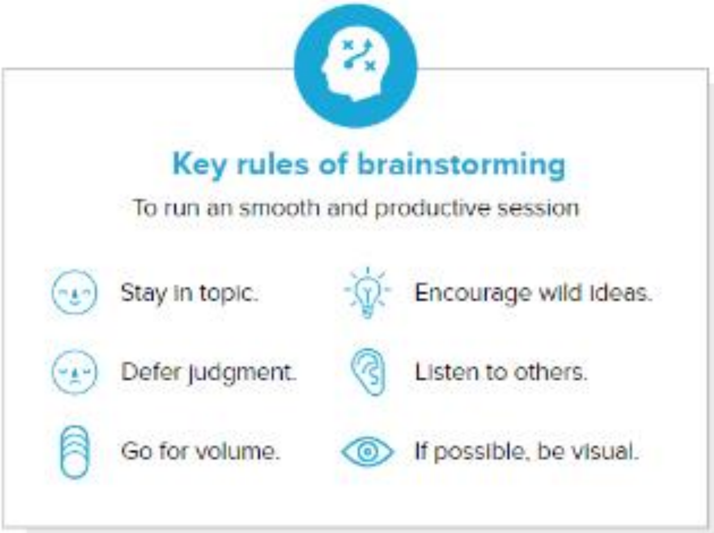
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

⏱ 5 minutes



PROBLEM

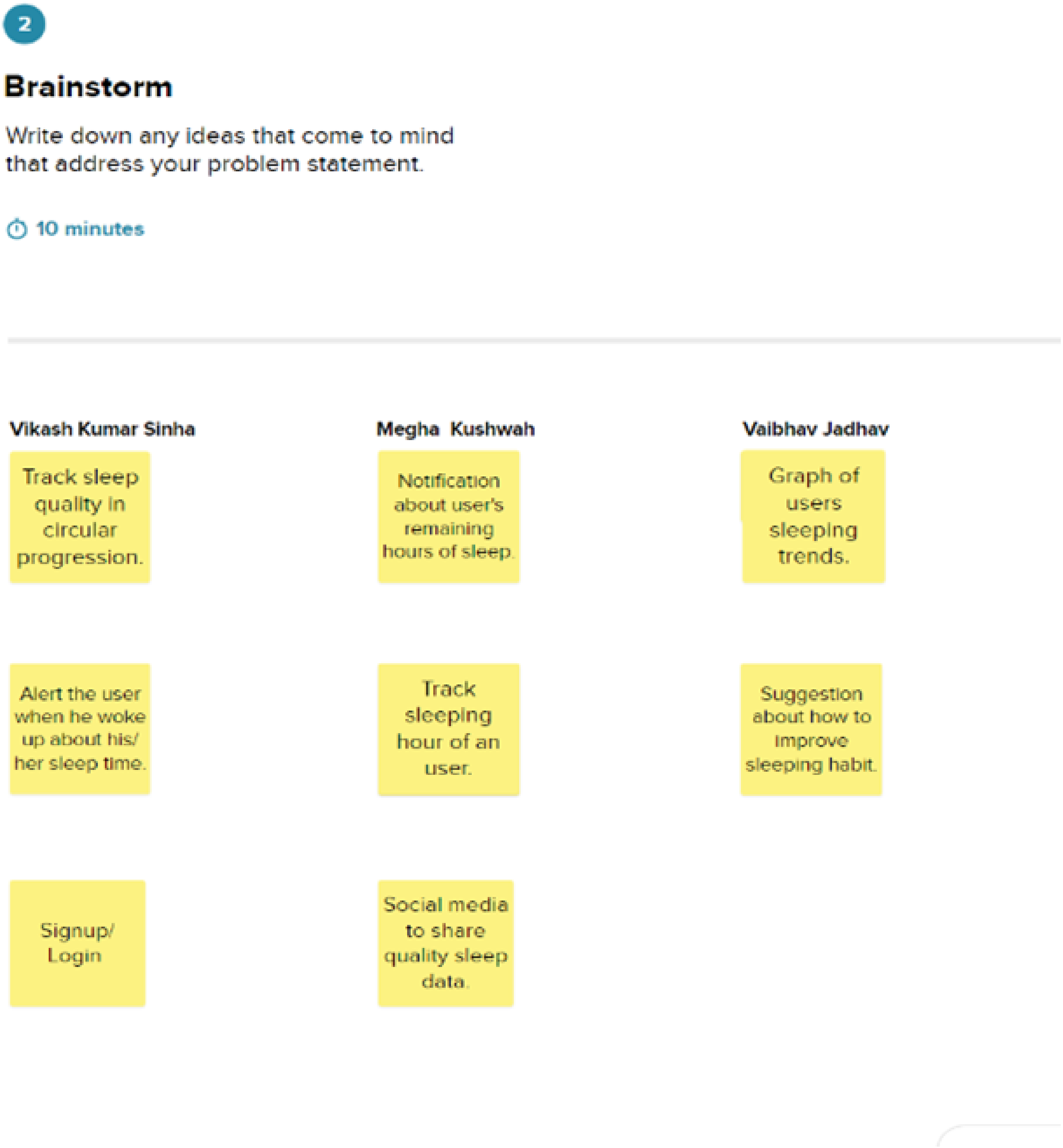
What features we are going to include in our sleep tracking app



Key rules of brainstorming

To run an smooth and productive session

Stay in topic.	Encourage wild ideas.
Defer judgment.	Listen to others.
Go for volume.	If possible, be visual.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Vikash Kumar Sinha

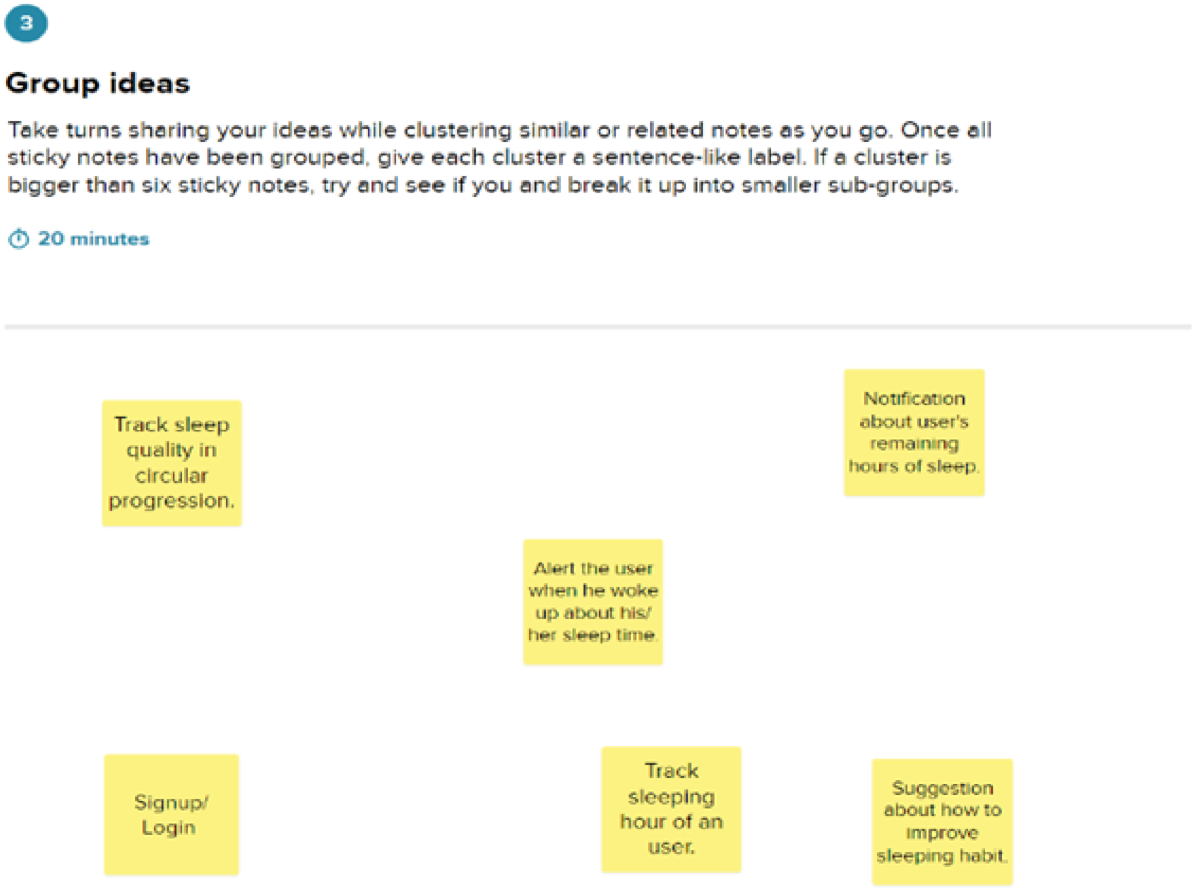
- Track sleep quality in circular progression.
- Alert the user when he woke up about his/ her sleep time.
- Signup/ Login

Megha Kushwah

- Notification about user's remaining hours of sleep.
- Track sleeping hour of an user.
- Social media to share quality sleep data.

Vaibhav Jadhav

- Graph of users sleeping trends.
- Suggestion about how to improve sleeping habit.



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Track sleep quality in circular progression.

Notification about user's remaining hours of sleep.

Alert the user when he woke up about his/ her sleep time.

Signup/ Login

Track sleeping hour of an user.

Suggestion about how to improve sleeping habit.

4

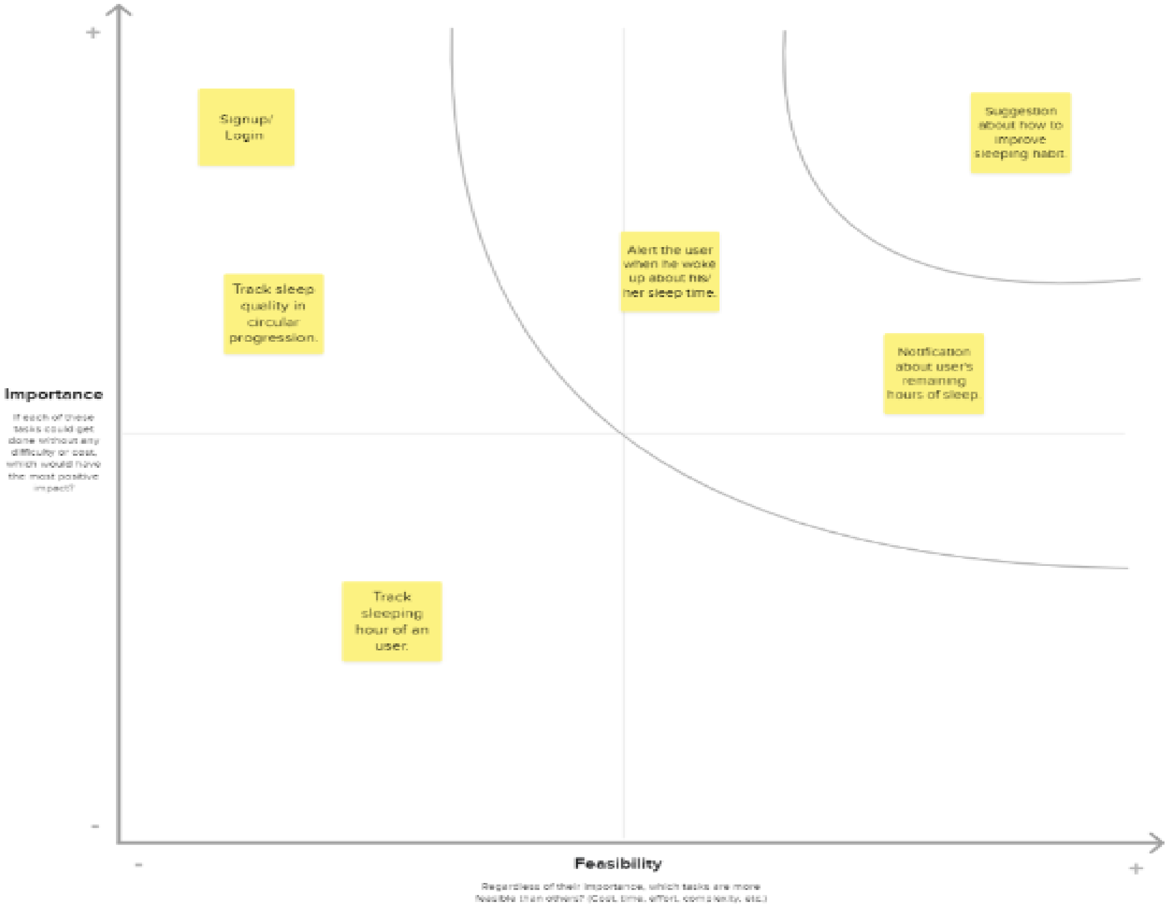
Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.



Signup/ Login

Track sleep quality in circular progression.

Track sleeping hour of an user.

Alert the user when he woke up about his/ her sleep time.

Suggestion about how to improve sleeping habit.

Notification about user's remaining hours of sleep.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

7. REQUIREMENT ANALYSIS

8. Functional Requirements

Functional requirements for a sleep tracking app should encompass a range of features that enable users to effectively monitor and improve their sleep patterns.

- Track sleep time
- Sleep quality checking
- Sleep analysis

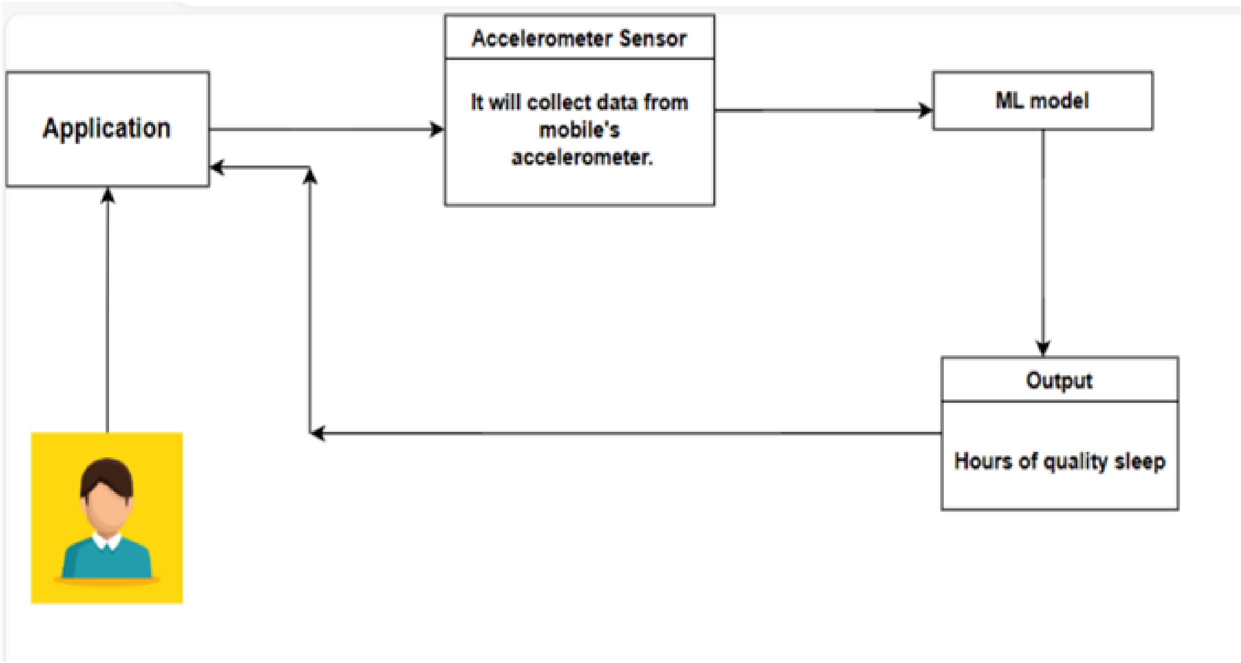
9. Non-Functional Requirements

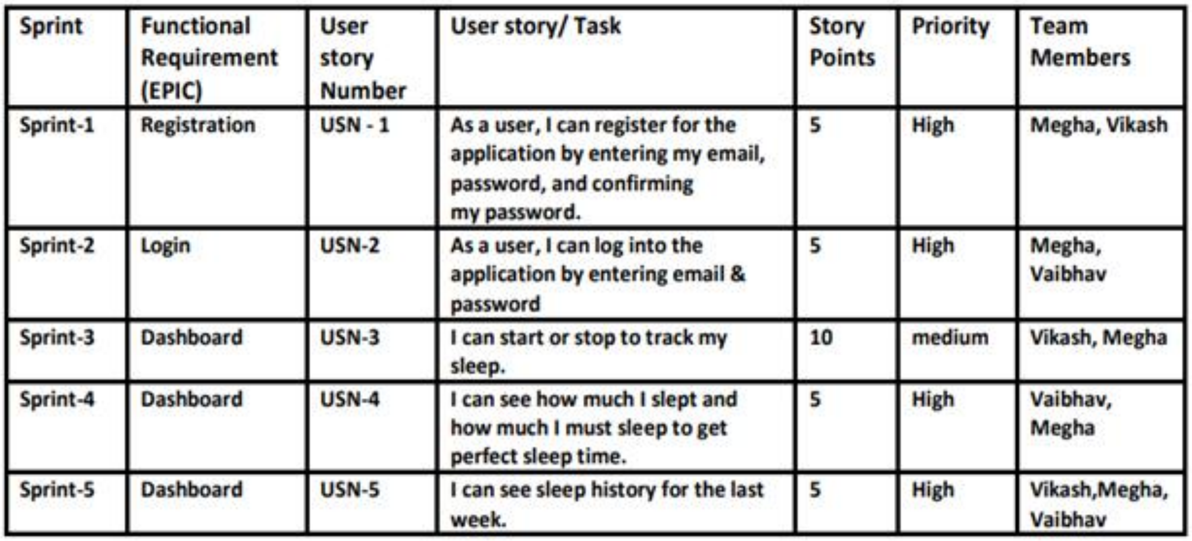
Non-functional requirements for a sleep tracking app outline the performance, reliability, and usability aspects of the app, ensuring it meets user expectations and functions effectively.

- Ease of Use
- Help and support
- Error Handling

10. PROJECT DESIGN

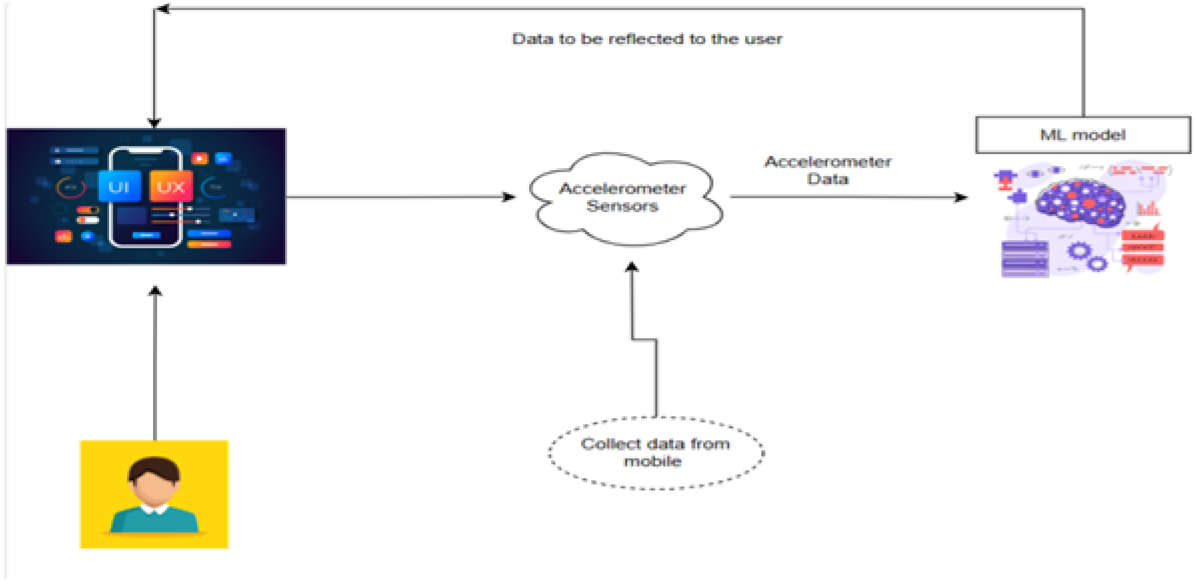
5.1 Data flow diagram and user stories





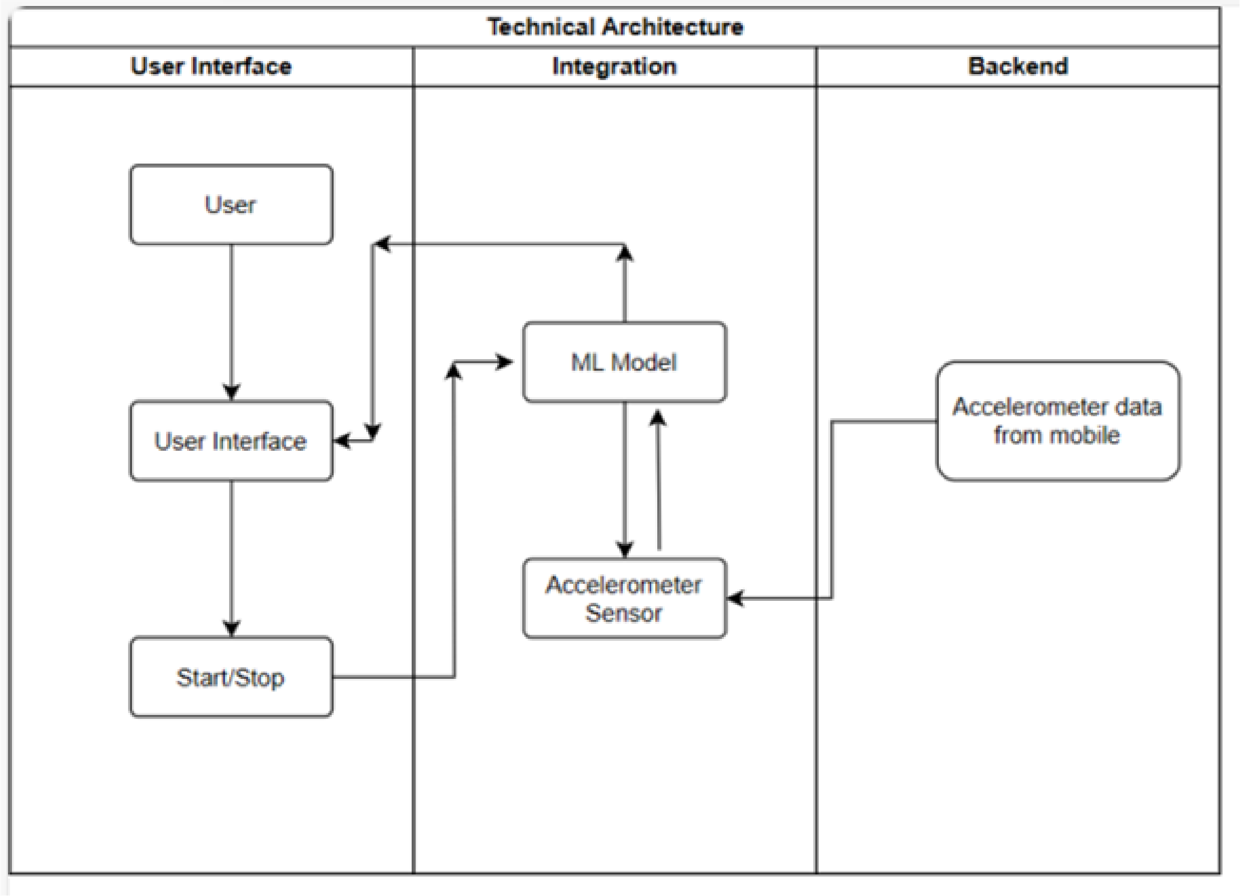
Sprint	Functional Requirement (EPIC)	User story Number	User story/ Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN - 1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Megha, Vikash
Sprint-2	Login	USN-2	As a user, I can log into the application by entering email & password	5	High	Megha, Vaibhav
Sprint-3	Dashboard	USN-3	I can start or stop to track my sleep.	10	medium	Vikash, Megha
Sprint-4	Dashboard	USN-4	I can see how much I slept and how much I must sleep to get perfect sleep time.	5	High	Vaibhav, Megha
Sprint-5	Dashboard	USN-5	I can see sleep history for the last week.	5	High	Vikash,Megha, Vaibhav

5.2 Solution Architecture

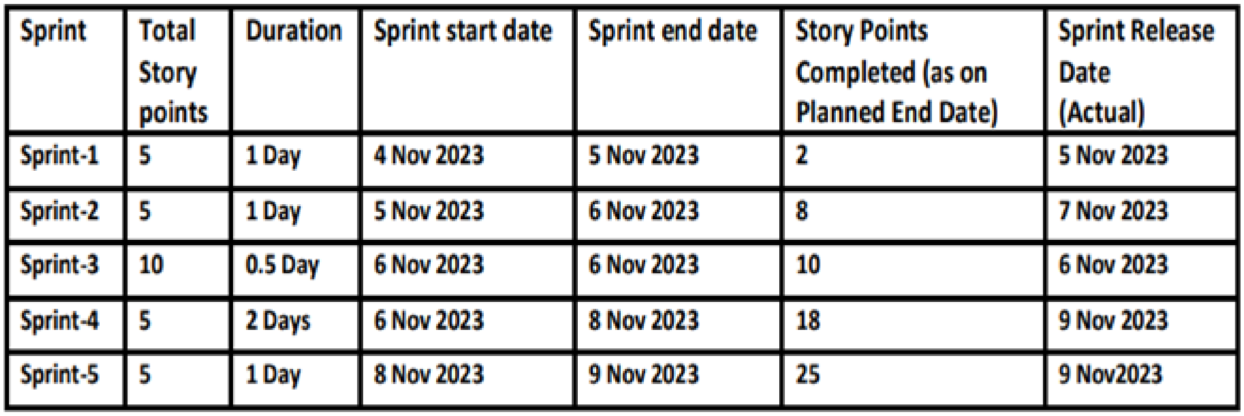


11. PROJECT PLANNING AND SCHEDULING

6.1 Technical Architecture


Technical Architecture
User Interface
Integration
Backend
User
User Interface
Start/Stop
ML Model
Accelerometer Sensor
Accelerometer data from mobile

12. Sprint planning and estimation



Sprint	Total Story points	Duration	Sprint start date	Sprint end date	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	5	1 Day	4 Nov 2023	5 Nov 2023	2	5 Nov 2023
Sprint-2	5	1 Day	5 Nov 2023	6 Nov 2023	8	7 Nov 2023
Sprint-3	10	0.5 Day	6 Nov 2023	6 Nov 2023	10	6 Nov 2023
Sprint-4	5	2 Days	6 Nov 2023	8 Nov 2023	18	9 Nov 2023
Sprint-5	5	1 Day	8 Nov 2023	9 Nov 2023	25	9 Nov2023

13. CODING & SOLUTIONING

7.1 Feature 1: User Registration

The Sleep Tracking App features a user registration process implemented in the RegistrationActivity. Users can provide their name, email, and password for registration. The app ensures all fields are filled in before allowing registration. Upon successful registration, users are redirected to the main activity (MainActivity).

7.2 Feature 1: User Login

The app includes a user login feature in the LoginActivity, allowing users to log in by entering their username and password. The app validates input, requiring non-empty fields for login. Successful login redirects users to the registration activity (RegistrationActivity), which may need adjustment for typical behavior. The app also provides a "Forget Password" feature for initiating password recovery.

7.3 Feature 3: Forget Password

This feature allows users to recover their password. Clicking on the "Forget Password" TextView navigates users to the ForgetPasswordActivity for further recovery steps.

7.4 Feature 4: Track Sleep Session

The TrackSleepActivity enables users to track their sleep sessions. Sleep session details, including start and end times, are passed through intents. The activity formats and displays these times in the startTimeTextView and endTimeTextView respectively.

7.5 Feature 5: Average Sleep Duration Calculation

The app could benefit from a feature to calculate the average sleep duration over multiple sessions. This would provide users with insights into their overall sleep patterns.

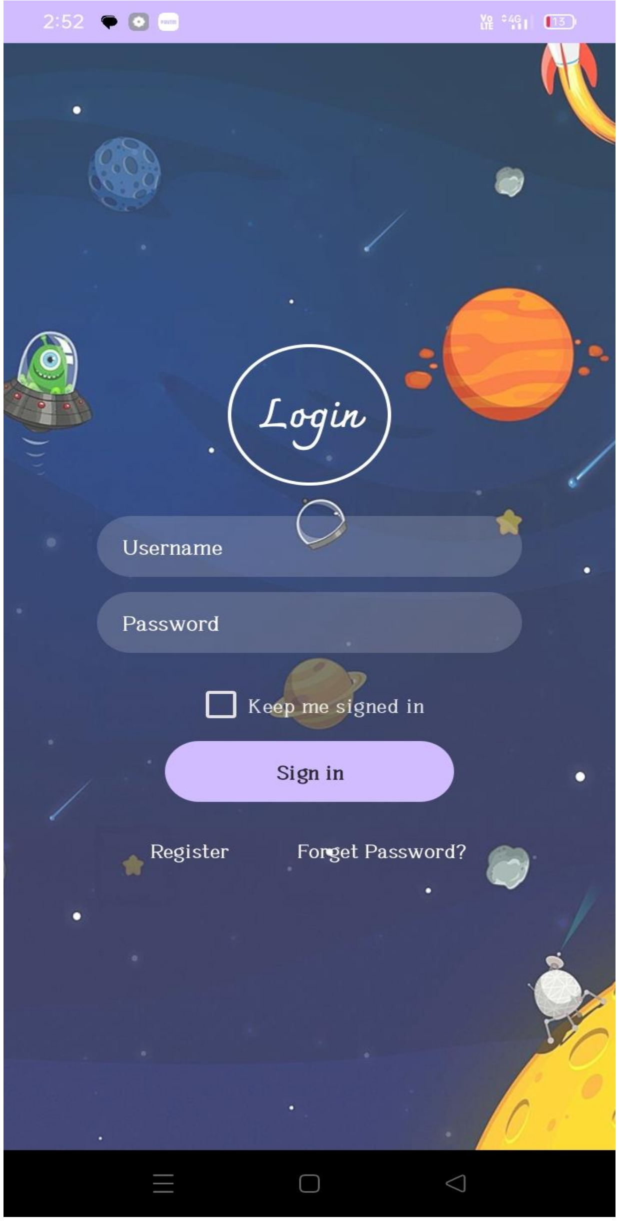
14. PERFORMANCE TESTING

8.1 Performance Metrics

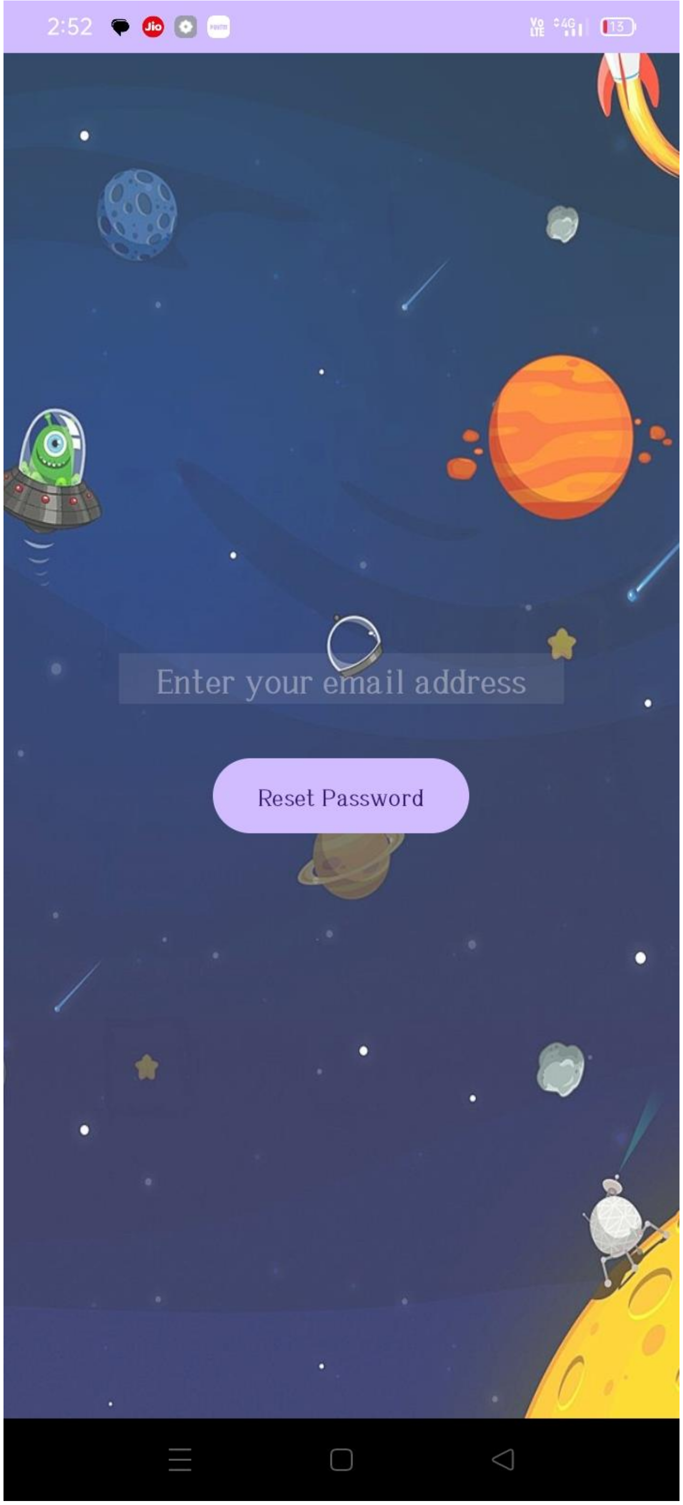
15. RESULTS

9.1 Output Screenshots

Final Output of the Application :
Login Page :



Reset Password Page:



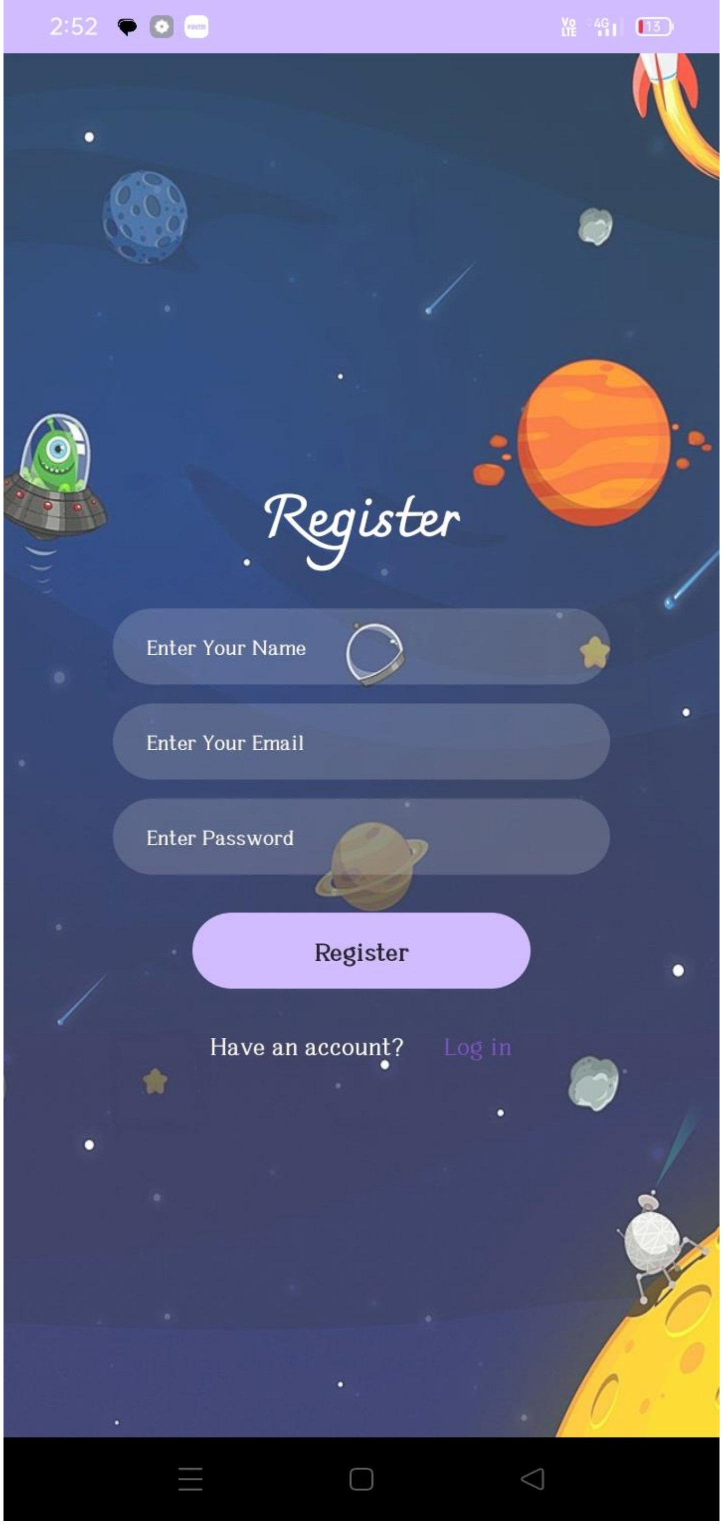
2:52

Enter your email address

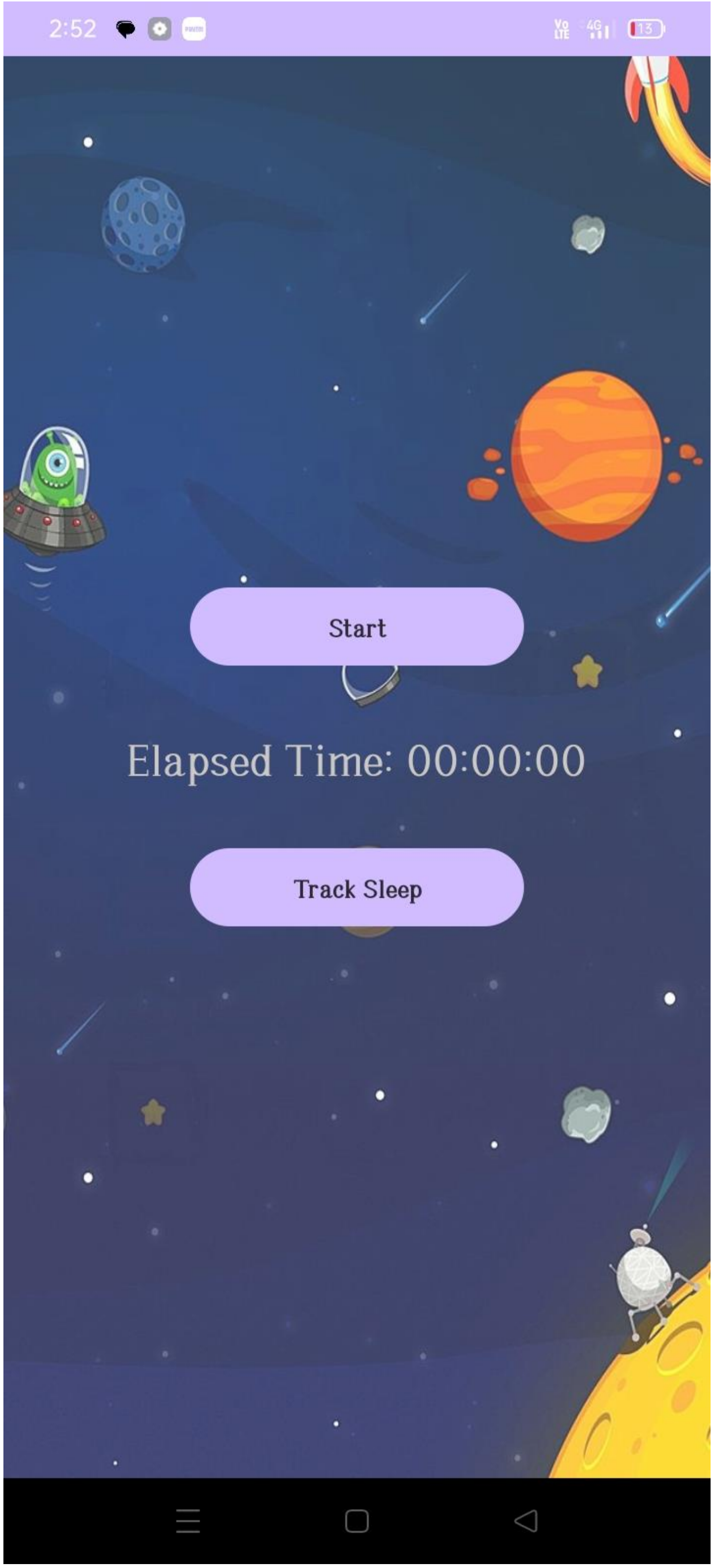
Reset Password

Registration Page :

Main Page:

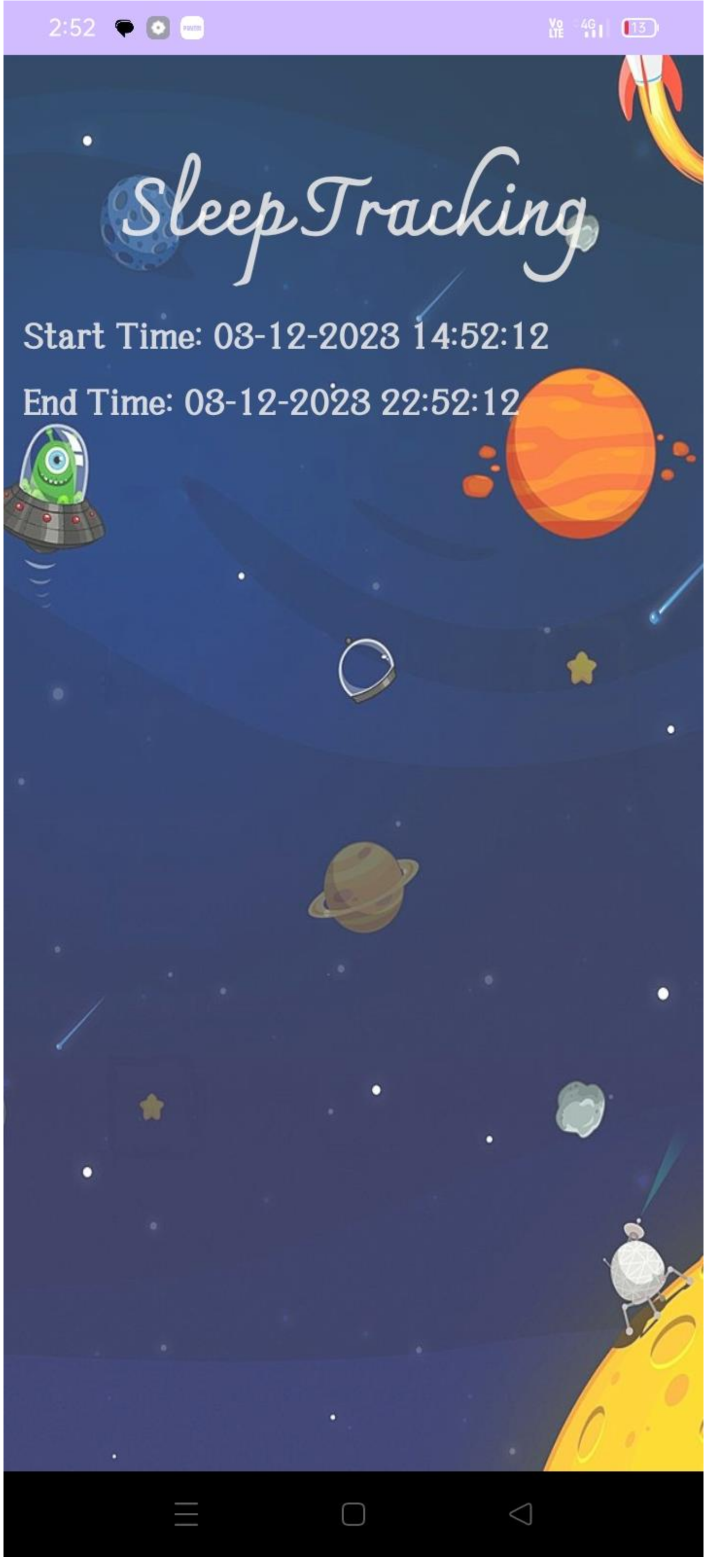


Main Page:



Elapsed Time: 00:00:00

Track Sleep Page:



Sleep Tracking

Start Time: 03-12-2023 14:52:12

End Time: 03-12-2023 22:52:12

16. ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES:

- Access to Sleep Information
- Figure Out Underlying Problems
- Learn How Interrupted Your Sleep Is
- Tracking can help you identify which areas of sleep need improvement
- Monitor Heart Rate
- Help Waking up

10.2 DISADVANTAGES

- Time Expenditure
- Overthinking
- Limited Feedback and Reliability
- The temptation to Check During the Night

17. CONCLUSION

The Sleep Tracking App project aims to address the need for better sleep quality by offering a user-friendly solution. The project identified existing issues with sleep tracking apps, proposed a comprehensive set of functional and non-functional requirements, and designed key features like user registration, login, and sleep session tracking. The results include output screenshots showcasing the app's interface and functionality.

While the app provides advantages such as easy access to sleep information and heart rate monitoring, it comes with potential drawbacks, including time expenditure and the temptation to check the app during the night. The success of the project lies in its ability to empower users to improve their sleep habits, acknowledging the balance between benefits and challenges in the pursuit of better overall well-being.

18. FUTURE SCOPE

19. Advanced Sleep Analysis Algorithms:

Invest in research and development to improve the accuracy of sleep analysis. Implement advanced algorithms that can better interpret sleep patterns, taking into account factors such as sleep cycles, stages, and interruptions.

20. Integration with Health Data:

Collaborate with health data platforms or APIs to integrate additional health-related information. This could include factors like physical activity, nutrition, or stress levels, providing users with a comprehensive view of their overall well-being.

21. Community and Social Features:

Introduce a community aspect to the app where users can share their sleep improvement journeys, tips, and challenges. Adding social features can enhance user engagement and create a supportive environment.

22. Smart Home Integration:

Explore integration with smart home devices to create an optimal sleep environment. This could involve connecting with smart lights, thermostats, or other devices that contribute to creating a conducive atmosphere for better sleep.

23. Real-time Sleep Tracking:

Develop real-time sleep tracking capabilities, allowing users to monitor their sleep patterns as they happen. This feature can provide instant feedback and may be particularly useful for those trying to implement immediate changes to their sleep habits.

APPENDIX
Source Code

24. Main Activity

```
package com.example.sleep_tracking_app

import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.FirebaseApp
import java.text.SimpleDateFormat
import java.util.Date
import java.util.Locale

class MainActivity : AppCompatActivity() {
    private lateinit var startSleepTrackingButton: Button
    private lateinit var elapsedTimeTextView: TextView
    private lateinit var trackSleepButton: Button

    private val handler = Handler(Looper.getMainLooper())
    private var startTimeMillis = 0L

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        FirebaseApp.initializeApp(this)

        setContentView(R.layout.activity_main)

        startSleepTrackingButton = findViewById(R.id.startSleepTrackingButton)
        elapsedTimeTextView = findViewById(R.id.elapsedTimeTextView)
        trackSleepButton = findViewById(R.id.trackSleepButton)

        startSleepTrackingButton.setOnClickListener {
            startTimeMillis = System.currentTimeMillis()
            handler.postDelayed(updateElapsedTime, 1000)
        }

        trackSleepButton.setOnClickListener {
            val intent = Intent(this, TrackSleepActivity::class.java)

            // Set start and end times as extras
            val currentTimeMillis = System.currentTimeMillis()
            val endTimeMillis = currentTimeMillis + (8 * 60 * 60 * 1000) // Assuming 8
hours of sleep

            intent.putExtra("startTime", currentTimeMillis)
            intent.putExtra("endTime", endTimeMillis)

            startActivity(intent)
        }
    }

    private val updateElapsedTime = object : Runnable {
        override fun run() {
            val currentTimeMillis = System.currentTimeMillis()
            val elapsedTimeMillis = currentTimeMillis - startTimeMillis

            val hours = (elapsedTimeMillis / (1000 * 60 * 60)) % 24
            val minutes = (elapsedTimeMillis / (1000 * 60)) % 60
            val seconds = (elapsedTimeMillis / 1000) % 60

            val formattedTime = String.format("%02d:%02d:%02d", hours, minutes, seconds)

            elapsedTimeTextView.text = "Elapsed Time: $formattedTime"

            handler.postDelayed(this, 1000)
        }
    }

    override fun onDestroy() {
        super.onDestroy()
        handler.removeCallbacks(updateElapsedTime)
    }
}
```

25. Login Activity

```
package com.example.sleep_tracking_app

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class LoginActivity : AppCompatActivity() {

    private lateinit var usernameEditText: EditText
    private lateinit var passwordEditText: EditText
    private lateinit var signInButton: Button
    private lateinit var forgetPasswordTextView: TextView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        usernameEditText = findViewById(R.id.usernameEditText)
        passwordEditText = findViewById(R.id.passwordEditText)
        signInButton = findViewById(R.id.signInButton)
        forgetPasswordTextView = findViewById(R.id.forgetPasswordTextView)

        signInButton.setOnClickListener {
            val username = usernameEditText.text.toString()
            val password = passwordEditText.text.toString()
            if (username.isNotEmpty() && password.isNotEmpty()) {
                // Valid username and password, open RegistrationActivity
                val intent = Intent(this, RegistrationActivity::class.java)
                startActivity(intent)
            } else {
                // Invalid username or password
                Toast.makeText(baseContext, "Could not sign in. Please try again.",
Toast.LENGTH_SHORT).show()
            }
        }

        forgetPasswordTextView.setOnClickListener {
            val intent = Intent(this, ForgetPasswordActivity::class.java)
            startActivity(intent)
        }
    }
}
```

26. Register Activity

```
package com.example.sleep_tracking_app

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class RegistrationActivity : AppCompatActivity() {

    private lateinit var nameEditText: EditText
    private lateinit var emailEditText: EditText
    private lateinit var passwordEditText: EditText
    private lateinit var registerButton: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_registration)

        nameEditText = findViewById(R.id.nameEditText)
        emailEditText = findViewById(R.id.emailEditText)
        passwordEditText = findViewById(R.id.passwordEditText)
        registerButton = findViewById(R.id.registerButton)

        registerButton.setOnClickListener {
            val name = nameEditText.text.toString()
            val email = emailEditText.text.toString()
            val password = passwordEditText.text.toString()
            if (name.isNotEmpty() && email.isNotEmpty() && password.isNotEmpty()) {
                // Valid input, open MainActivity
                val intent = Intent(this, MainActivity::class.java)
                startActivity(intent)
                finish()
            } else {
                // Invalid input
                Toast.makeText(baseContext, "Please fill in all fields.",
Toast.LENGTH_SHORT).show()
            }
        }
    }S
}
```

27.ForgotPassword Activity

```
package com.example.sleep_tracking_app

import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.auth.FirebaseAuth

class ForgetPasswordActivity : AppCompatActivity() {

    private lateinit var emailEditText: EditText
    private lateinit var resetButton: Button

    private lateinit var auth: FirebaseAuth

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_forget_password)

        auth = FirebaseAuth.getInstance()

        emailEditText = findViewById(R.id.emailEditText)
        resetButton = findViewById(R.id.resetButton)

        resetButton.setOnClickListener {
            val email = emailEditText.text.toString()

            auth.sendPasswordResetEmail(email)
                .addOnCompleteListener { task ->
                    if (task.isSuccessful) {
                        // Password reset email sent successfully
                        Toast.makeText(this, "Password reset email sent!",
Toast.LENGTH_SHORT).show()
                        finish()
                    } else {
                        // If sending email fails, display a message to the user.
                        Toast.makeText(this, "Password reset failed. Please try again.",
Toast.LENGTH_SHORT).show()
                    }
                }
        }
    }
}
```

28. TrackSleep Activity

```
package com.example.sleep_tracking_app

import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import java.text.SimpleDateFormat
import java.util.Date
import java.util.Locale

class TrackSleepActivity : AppCompatActivity() {

    private lateinit var startTimeTextView: TextView
    private lateinit var endTimeTextView: TextView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_track_sleep)

        startTimeTextView = findViewById(R.id.startTimeTextView)
        endTimeTextView = findViewById(R.id.endTimeTextView)

        val startTimeMillis = intent.getLongExtra("startTime", 0)
        val endTimeMillis = intent.getLongExtra("endTime", 0)

        val startTime = formatTime(startTimeMillis)
        val endTime = formatTime(endTimeMillis)

        startTimeTextView.text = "Start Time: $startTime"
        endTimeTextView.text = "End Time: $endTime"
    }

    private fun formatTime(timeMillis: Long): String {
        val dateFormat = SimpleDateFormat("dd-MM-yyyy", Locale.getDefault())
        val timeFormat = SimpleDateFormat("HH:mm:ss", Locale.getDefault())

        val date = Date(timeMillis)
        val formattedDate = dateFormat.format(date)
        val formattedTime = timeFormat.format(date)

        return "$formattedDate $formattedTime"
    }
}
```

29. Manifest.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"
    package="com.example.sleep_tracking_app">

    <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:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Sleep_tracking_app"
        tools:targetApi="31">

        <activity
            android:name=".ForgetPasswordActivity"
            android:exported="false" />

        <activity
            android:name=".TrackSleepActivity"
            android:exported="false" />

        <activity
            android:name=".LoginActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".RegistrationActivity"
            android:exported="false" />

        <activity
            android:name=".MainActivity"
            android:exported="false" />

        <meta-data
            android:name="preloaded_fonts"
            android:resource="@array/preloaded_fonts" />
    </application>

</manifest>
```

30.Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
</resources>
```

31.String.xml

```
<resources>
    <string name="app_name">Sleep_tracking_app</string>
</resources>
```

Git Hub Link: https://github.com/vikash-kumar-sinha/Sleep_tracking_app

Project Demo Link: https://drive.google.com/file/d/1kXlQBAN1un-X6SWnjTTzz7bOiCNcXB8u/view?usp=sharing