## **CALORIE - A SMART TRACKING APP**

## **A MINI PROJECT REPORT**

*Submitted by*

**MITHILESH T 220701165**

*in partial fulfillment for the course*

**CS19611 – MOBILE APPLICATION DEVELOPMENT LABORATORY**

*of the degree of*

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**

|  |  |
| --- | --- |
|  | C:\Users\SPS\Downloads\REC-Logo-Autnonomous.jpg |

**RAJALAKSHMI ENGINEERING COLLEGE**

**RAJALAKSHMI NAGAR**

**THANDALAM,CHENNAI – 602 105**

**MAY 2025**

**BONAFIDE CERTIFICATE**

Certified that this project report “**CALORIE**” is the bonafide work of

**“ MITHILESH T (220701165) ”**

who carried out the project work under my supervision.

**Submitted for the Practical Examination held on**

**SIGNATURE**

**MrV..KARTHICK**

Assistant Professor(SG)

Computer Science and Engineering

Rajalakshmi Engineering College

Thandalam, Chennai-602 105

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

I express my sincere thanks to my beloved and honorable chairman **MR. S.**

**MEGANATHAN** and the chairperson **DR.M.THANGAM MEGANATHAN** for their timely support and encouragement.

I am greatly indebted to my respected and honorable principal **Dr. S.N**

**.MURUGESAN** for his able support and guidance. No words of gratitude will suffice for the unquestioning support extended to us by my head of the department **Dr. P. KUMAR**, and my Academic Head **Dr.N.DURAIMURUGAN**, for being ever supporting force during my project work.

I also extend my sincere and hearty thanks to my internal guide **Mr.V.KARTHICK** for his valuable guidance and motivation during the completion of this project.

My sincere thanks to my family members, friends and other staff members of Computer Science and Engineering.

**MITHILESH T 2116220701165**

# 

# 

# **ABSTRACT**

QuickScribe is an intuitive Android note-taking application developed using Kotlin and Android Studio, designed to offer users a seamless experience for creating, editing, and managing their notes efficiently. The app leverages the power of the Room Persistence Library to store all notes locally, ensuring fast access and reliable data persistence without requiring an internet connection. With a clean and user-friendly interface built following Material Design guidelines, QuickScribe allows users to quickly add new notes with minimal effort, edit existing notes with real-time updates, and delete unwanted notes permanently. The application follows modern Android development best practices, utilizing the Model-View-ViewModel (MVVM) architecture pattern to separate concerns and maintain a clean codebase. LiveData and Coroutines are employed to handle asynchronous operations and ensure smooth performance, while RecyclerView efficiently displays the list of notes. Designed with simplicity in mind, QuickScribe focuses on core functionality without unnecessary features, making it ideal for users who need a straightforward and reliable tool for their daily note-taking needs. The app's lightweight design and efficient use of system resources ensure optimal performance even on lower-end devices, while its offline capability guarantees accessibility anytime, anywhere. Whether for personal reminders, quick ideas, or important notes, QuickScribe provides a dependable solution that prioritizes speed, simplicity, and usability, making it a practical choice for anyone looking to streamline their note-taking process on Android devices.

**TABLE OF CONTENTS**

**1. INTRODUCTION**

1.1 IMPLEMENTATION

**2. SYSTEM SPECIFICATION**

2.1 HARDWARE SPECIFICATION

2.2 SOFTWARE SPECIFICATION

**3. SOURCE CODE**

**4. SNAPSHOTS**

**5. CONCLUSION**

**6.REFERENCES**

**CHAPTER 1**

**1. INTRODUCTION:**

#### In the age of health-conscious living, personalized fitness tools are essential for maintaining well-being. *CalorieMate* addresses this demand by offering a streamlined, user-centric Android application designed to calculate an individual’s daily calorie requirements based on core health metrics like age, weight, height, and activity level. Developed using Kotlin and modern Android architecture components, the app ensures a seamless user experience with accurate and responsive calculations. Utilizing local computation and a lightweight design, CalorieMate provides quick access to essential nutritional insights without requiring an internet connection. Its clean, intuitive interface supports users in making informed dietary decisions, making it an ideal companion for both casual health trackers and fitness enthusiasts.

#### **1.1 IMPLEMENTATION**

Calorie employs a streamlined technical stack:

* **Frontend**: Built with Android Jetpack components using Kotlin
* **Architecture**: Model-View-ViewModel (MVVM) pattern
* **Database**: Room Persistence Library with SQLite backend
* **UI Components**:
  + **Structured Input Fields**: Uses EditText, RadioGroup, and Spinner to collect age, weight, height, gender, and activity level effectively.
  + **Responsive Layout**: Built with ScrollView and LinearLayout to ensure a smooth experience across different screen sizes.
* **Key Features Implementation**:
  + **Personalized Calorie Calculation**: Implements BMR-based logic tailored by gender and activity level to estimate daily calorie needs.
  + **Real-time Validation and Output**: Validates user inputs and displays formatted results instantly, enhancing usability and accuracy.

**CHAPTER 2**

### **2. SYSTEM SPECIFICATION**

### The following specifications outline the technical requirements for both developing and running the *CalorieMate* application. The hardware specifications define the minimum device capabilities needed for smooth functionality, while the software specifications describe the development environment and technologies used.

### **2.1 HARDWARE SPECIFICATION**

|  |  |
| --- | --- |
| **Component** | **Minimum Requirement** |
| Processor | ARMv8 64-bit or x86\_64 |
| RAM | 2GB (4GB recommended) |
| Storage | 50MB free space |
| Display | 5" HD (720×1280) |

#### 

#### 

#### **2.2 SOFTWARE SPECIFICATION**

|  |  |
| --- | --- |
| **Component** | **Version/Technology** |
| Operating System | Android 8.0 (Oreo) or later |
| Development IDE | Android Studio Giraffe 2022.3.1+ |
| Language | Kotlin 1.8+ |
| SDK Tools | Android SDK 33+ |
| Database | Room 2.5.2 |
| Architecture | MVVM with Clean Architecture |

**CHAPTER 3**

**Source Code**

**Activity\_main.xml**

<?xmlversion="1.0"encoding="utf-8"?>  
<ScrollViewxmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="20dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center\_horizontal">  
  
 <TextView  
 android:text="Daily Calorie Needs Calculator"  
 android:textSize="22sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="20dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
  
 <EditText  
 android:id="@+id/ageInput"  
 android:hint="Age (years)"  
 android:inputType="number"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <EditText  
 android:id="@+id/weightInput"  
 android:hint="Weight (kg)"  
 android:inputType="numberDecimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"/>  
  
 <EditText  
 android:id="@+id/heightInput"  
 android:hint="Height (cm)"  
 android:inputType="numberDecimal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"/>  
  
 <RadioGroup  
 android:id="@+id/genderGroup"  
 android:orientation="horizontal"  
 android:layout\_marginTop="10dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content">  
  
 <RadioButton  
 android:id="@+id/maleRadio"  
 android:text="Male"  
 android:checked="true"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
  
 <RadioButton  
 android:id="@+id/femaleRadio"  
 android:text="Female"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp" />  
 </RadioGroup>  
  
 <Spinner  
 android:id="@+id/activitySpinner"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp" />  
  
 <Button  
 android:id="@+id/calculateButton"  
 android:text="Calculate"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="15dp" />  
  
 <TextView  
 android:id="@+id/resultText"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:textColor="#333"  
 android:layout\_marginTop="20dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
  
 </LinearLayout>  
</ScrollView>

**AndroidM.xml**

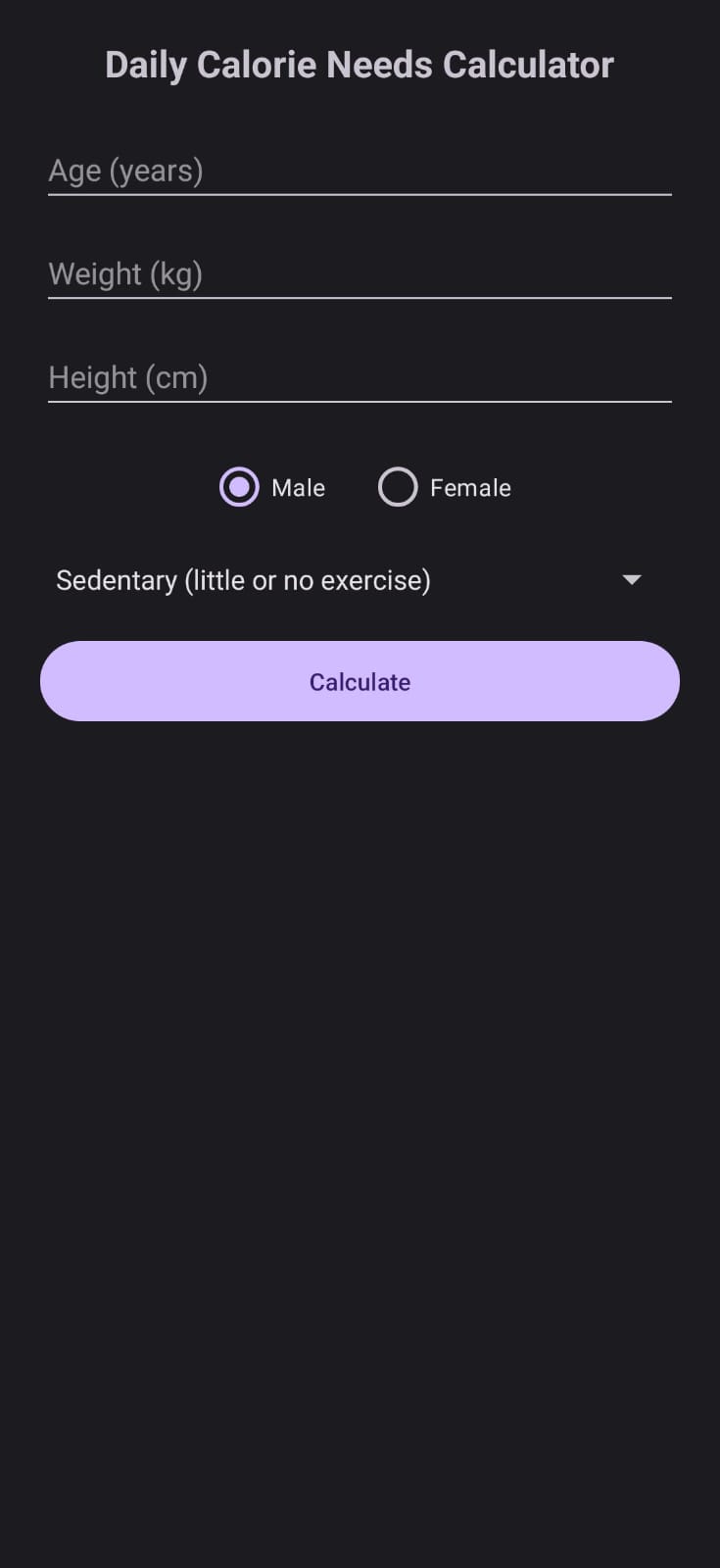
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Calorie"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

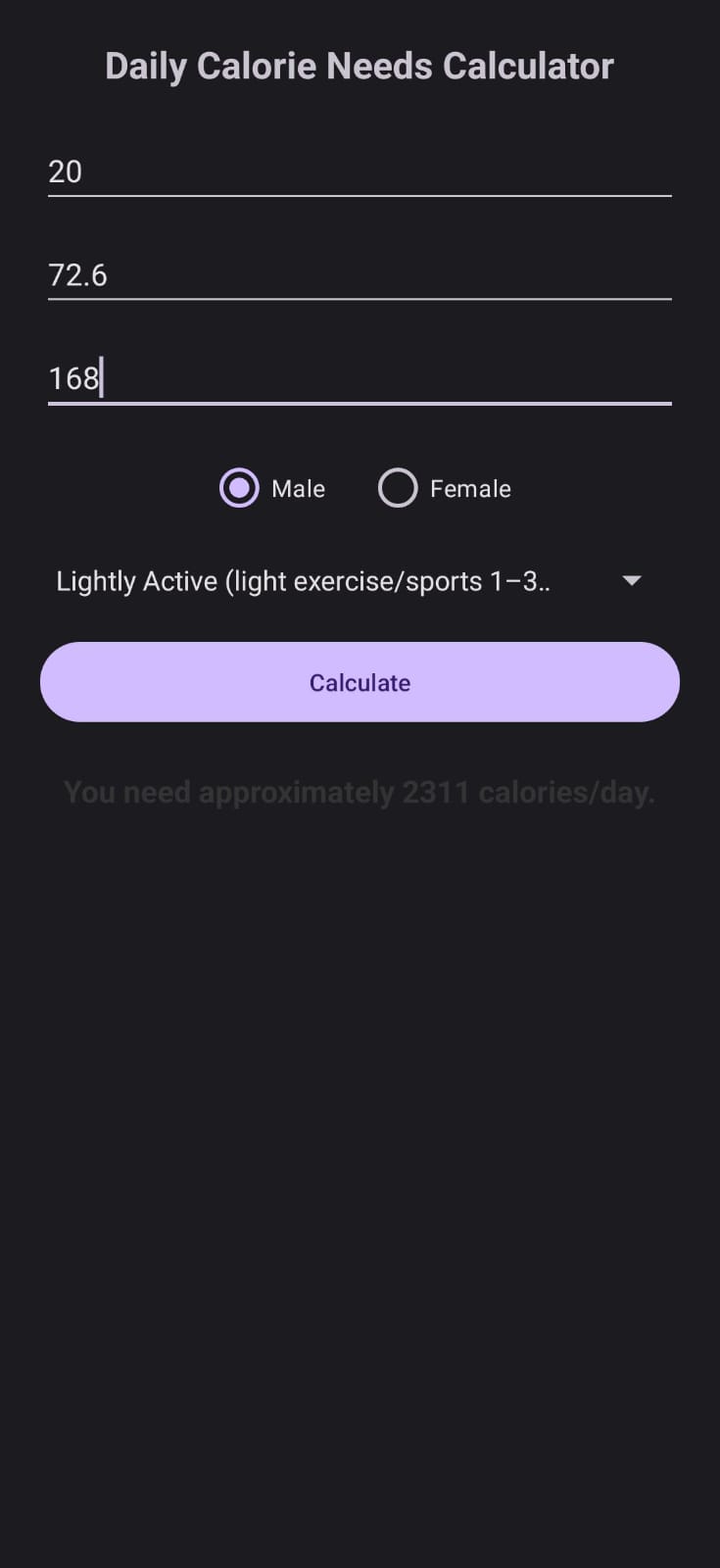
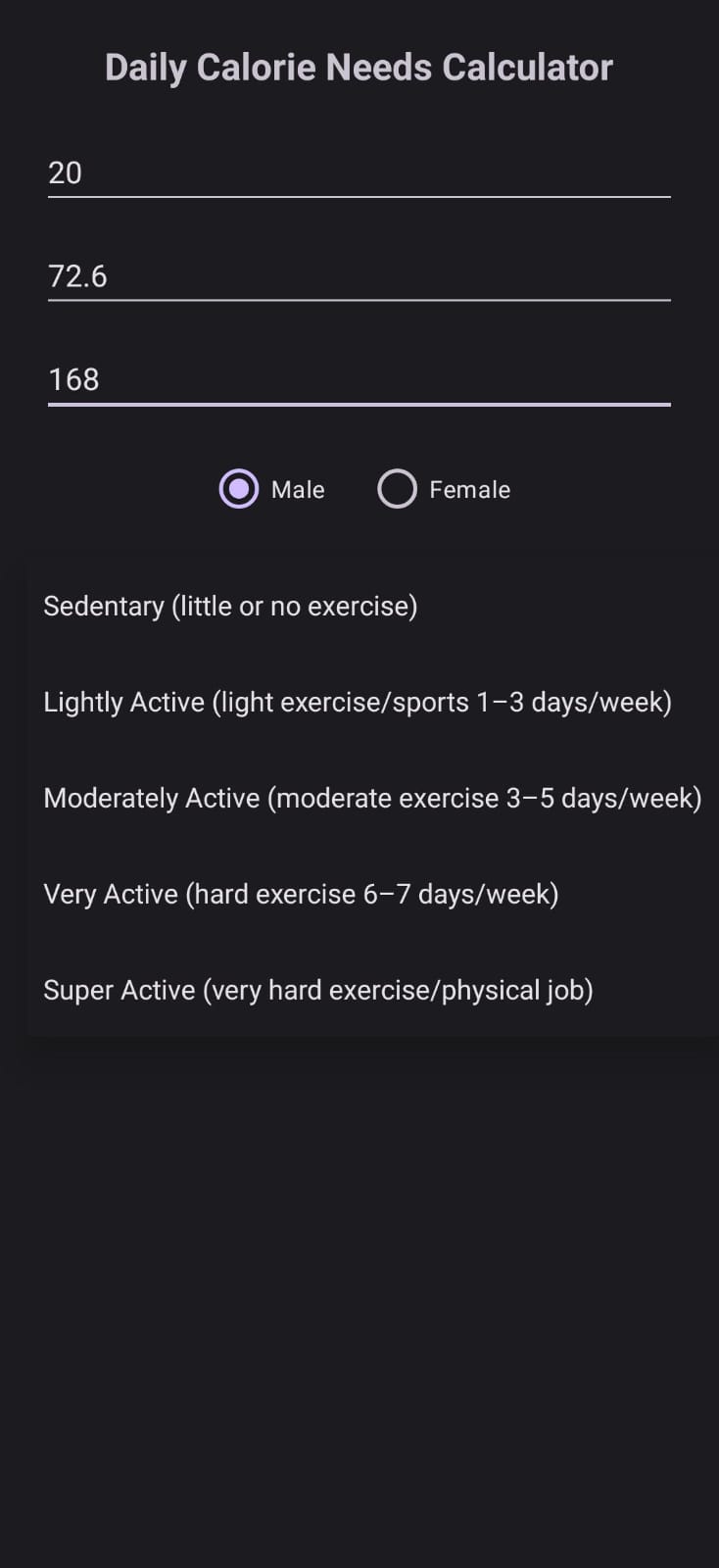
**MainActivity.kt**

package com.example.calorie  
  
import android.os.Bundle  
import android.widget.\*  
import androidx.appcompat.app.AppCompatActivity  
  
class MainActivity : AppCompatActivity() {  
  
 private lateinit var ageInput: EditText  
 private lateinit var weightInput: EditText  
 private lateinit var heightInput: EditText  
 private lateinit var genderGroup: RadioGroup  
 private lateinit var activitySpinner: Spinner  
 private lateinit var calculateButton: Button  
 private lateinit var resultText: TextView  
  
 private val activityLevels = mapOf(  
 "Sedentary (little or no exercise)" to 1.2,  
 "Lightly Active (light exercise/sports 1–3 days/week)" to 1.375,  
 "Moderately Active (moderate exercise 3–5 days/week)" to 1.55,  
 "Very Active (hard exercise 6–7 days/week)" to 1.725,  
 "Super Active (very hard exercise/physical job)" to 1.9  
 )  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_main)  
  
 ageInput = findViewById(R.id.ageInput)  
 weightInput = findViewById(R.id.weightInput)  
 heightInput = findViewById(R.id.heightInput)  
 genderGroup = findViewById(R.id.genderGroup)  
 activitySpinner = findViewById(R.id.activitySpinner)  
 calculateButton = findViewById(R.id.calculateButton)  
 resultText = findViewById(R.id.resultText)  
  
 // Populate Spinner  
 val adapter = ArrayAdapter(this, android.R.layout.simple\_spinner\_item, activityLevels.keys.toList())  
 adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item)  
 activitySpinner.adapter = adapter  
  
 calculateButton.setOnClickListener **{** calculateCalories()  
 **}** }  
  
 private fun calculateCalories() {  
 val age = ageInput.text.toString().toIntOrNull()  
 val weight = weightInput.text.toString().toDoubleOrNull()  
 val height = heightInput.text.toString().toDoubleOrNull()  
  
 if (age == null || weight == null || height == null) {  
 Toast.makeText(this, "Please fill in all fields", Toast.LENGTH\_SHORT).show()  
 return  
 }  
  
 val gender = when (genderGroup.checkedRadioButtonId) {  
 R.id.maleRadio -> "male"  
 R.id.femaleRadio -> "female"  
 else -> "male"  
 }  
  
 val bmr = if (gender == "male") {  
 10 \* weight + 6.25 \* height - 5 \* age + 5  
 } else {  
 10 \* weight + 6.25 \* height - 5 \* age - 161  
 }  
  
 val activityFactor = activityLevels[activitySpinner.selectedItem.toString()] ?: 1.2  
 val dailyCalories = bmr \* activityFactor  
  
 resultText.text = "You need approximately %.0f calories/day.".format(dailyCalories)}}

**CHAPTER 4**

**SNAP SHOTS OF QUICKSCRIBE**





**CHAPTER 5**

**CONCLUSION**

The *CalorieMate* Android application effectively demonstrates how modern mobile development tools and practices can be used to deliver a focused, health-oriented utility. Built using Kotlin and structured around the MVVM architecture, the app provides users with a fast and reliable way to calculate their daily calorie requirements based on age, weight, height, gender, and activity level. By leveraging local processing and a responsive Material Design interface, the application ensures accessibility and ease of use without relying on internet connectivity.

This project showcases the practical integration of Android Jetpack components to maintain a clean and maintainable codebase. The use of ViewModel promotes separation of concerns, while the intuitive UI offers real-time feedback and input validation. Though currently centered on calorie computation, the architecture is scalable, allowing for future enhancements such as dietary planning, macro tracking, or integration with wearable health devices. *CalorieMate* serves both as a helpful wellness companion for users and as an educational reference for implementing efficient, user-centric Android applications.