Project Report

(July 2025 – November 2025)

on

CSE226 : ANDROID APP DEPLOYMENT

Project Tittle: Smart Reminder App

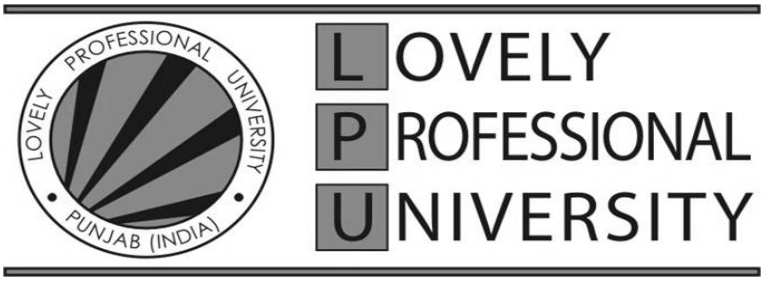
Submitted by

Pooskuri Rishi	**Registration Number : 12220397**
Mandapati Lokesh	**Registration Number : 12222069**
Nikhil Sai .k	**Registration Number : 12222121**

Under the Guidance of

Prof. Vikas Sharma

School of Computer Science and Engineering



Description of Project :

The **Smart Reminder Android Application** is a practical and user-friendly mobile solution designed to help individuals take their medicines on time. It simplifies daily medication management by providing timely alerts, organized schedules, and an easy-to-navigate interface. The app focuses on improving user health, convenience, and consistency while showcasing strong Android development skills using **Kotlin**, **SQLite**, and **AlarmManager**.

Key Features of the System

1. User Authentication

- Splash screen and interactive welcome page
- Simple login and signup options
- Secure local validation of user details
- Smooth navigation to the home dashboard after login

2. Medicine Management

- Option to **add new medicines** with name, dosage, frequency, and time
- Displays all added medicines using a RecyclerView/Adapter view
- Progress indicator showing **completed vs pending** medicines
- Easy editing or removal of medicine entries

3. Smart Reminder System

- Uses **AlarmManager** to schedule medicine reminders
- Sends **timely notifications/alerts** for taking medicines
- Works even when the app is closed
- Ensures users never miss important medication timings

4. History Logs

- Stores and displays **past medication records**
- Shows which medicines were taken or missed
- Helps users monitor their medication habits over time

5. Emergency Call Feature

- One-tap **emergency call** button for quick access
- Helps users reach assistance immediately when needed

Code

1. Manifes.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">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS"
/>
    <uses-permission
android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission
android:name="android.permission.SCHEDULE_EXACT_ALARM" />
    <uses-permission android:name="android.permission.USE_EXACT_ALARM" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/logo"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/logo_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SmartRemainder">
        <activity
            android:name=".data_anaysis"
            android:exported="false" />
        <activity
            android:name=".AddMedicineActivity"
            android:exported="false" />
        <activity
            android:name=".CalendarActivity"
            android:exported="false" />
        <activity
            android:name=".HistoryActivity"
            android:exported="false" />
        <activity
            android:name=".RefillActivity"
            android:exported="false" />
        <activity
            android:name=".ProfileActivity"
            android:exported="false" />
```

```
        <activity
            android:name=".Splashscreen"
            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=".Signukt"
            android:exported="true" />
        <activity
            android:name=".WelComePage"
            android:exported="true" />
        <activity
            android:name=".Login"
            android:exported="true" />
        <activity
            android:name=".MainActivity"
            android:exported="true" />

        <receiver
            android:name=".AlarmReceiver"
            android:enabled="true"
            android:exported="false" />
        <receiver
            android:name=".BootReceiver"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.BOOT_COMPLETED" />
            </intent-filter>
        </receiver>

        <service android:name=".RingtoneService" />

    </application>
    <queries>
        <intent>
            <action android:name="android.intent.action.GET_CONTENT" />
            <data android:mimeType="image/*" />
        </intent>
    </queries>

</manifest>
```

2. Gradle

```
import java.util.Properties

plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
}

android {
    namespace = "com.example.smartremainder"
    compileSdk = 36

    defaultConfig {
        applicationId = "com.example.smartremainder"
        minSdk = 24
        targetSdk = 36
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"

    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
    kotlinOptions {
        jvmTarget = "11"
    }
    buildFeatures {
        viewBinding = true
        buildConfig = true // 3. Enable BuildConfig
    }
    // Corrected aaptOptions block with proper Kotlin DSL syntax
```

```
    aaptOptions {
        noCompress.add("tflite")
    }
}

dependencies {

    implementation(libs.androidx.core.ktx)
    implementation(libs.androidx.appcompat)
    implementation(libs.material)
    implementation(libs.androidx.activity)
    implementation(libs.androidx.constraintlayout)
    implementation("com.github.bumptech.glide:glide:4.12.0")
    annotationProcessor("com.github.bumptech.glide:compiler:4.12.0")
    implementation("com.github.PhilJay:MPAndroidChart:v3.1.0")
    implementation("com.squareup.okhttp3:okhttp:4.12.0")
    // Simplified and corrected TensorFlow Lite dependencies
    implementation("org.tensorflow:tensorflow-lite:2.17.0")
    implementation("org.tensorflow:tensorflow-lite-support:0.5.0")
    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
}
```

3. Color / string / theme

```
<?xml version="1.0" encoding="utf-8"?>

<resources>

    <color name="purple_200">#FFBB86FC</color>

    <color name="purple_500">#FF6200EE</color>

    <color name="purple_700">#FF3700B3</color>

    <color name="teal_200">#FF03DAC5</color>

    <color name="teal_700">#FF018786</color>

    <color name="black">#FF000000</color>

    <color name="white">#FFFFFFFF</color>

    <color name="text_secondary">#8A000000</color>

    <color name="green_primary">#4CAF50</color>

    <color name="green_dark">#388E3C</color>
```

```
    <color name="green_light">#C8E6C9</color>
    <color name="grey_light">#D3D3D3</color>
    <color name="button_unselected">#E0E0E0</color>
    <color name="text_primary">#DE000000</color>
    <color name="dark_gray">#A9A9A9</color>
    <color name="chart_bar_background">#F0F0F0</color>
    <color name="chart_blue">#5C6BC0</color>
    <color name="chart_green">#66BB6A</color>
    <color name="chart_red">#EF5350</color>
    <color name="chart_yellow">#FFEE58</color>
    <color name="blue_primary">#3949AB</color>
    <color name="blue_dark">#303F9F</color>
    <color name="red_dark">#D32F2F</color>
</resources>
```

4. Xml code

Activity_main.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F5F5F5">

    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fillViewport="true">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:padding="16dp">
```

```
<!-- ✅ User Info & Emergency Call -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:orientation="horizontal"
    android:padding="8dp"
    android:layout_marginBottom="8dp">

    <!-- Profile Image -->
    <ImageView
        android:id="@+id/ivProfile"
        android:layout_width="55dp"
        android:layout_height="55dp"
        android:layout_marginEnd="10dp"
        android:background="@drawable/profile_circle_bg"
        android:contentDescription=""
        android:padding="8dp"
        android:src="@drawable/ic_user_profile" />

    <!-- User Info -->
    <TextView
        android:id="@+id/tvUserName"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Welcome, User!"
        android:textColor="#333333"
        android:textSize="18sp"
        android:textStyle="bold" />

    <!-- Emergency Call -->
    <LinearLayout
        android:id="@+id/llEmergencyCall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/bg_emergency_call"
        android:clickable="true"
        android:focusable="true"
        android:gravity="center"
        android:orientation="horizontal"
        android:padding="8dp">

        <ImageView
            android:layout_width="22dp"
            android:layout_height="22dp"
```

```
            android:src="@android:drawable/ic_menu_call"
            app:tint="@android:color/white" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="4dp"
            android:text="Emergency"
            android:textColor="@android:color/white"
            android:textStyle="bold" />
    </LinearLayout>
</LinearLayout>

<!-- ✅ Progress Section -->
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="230dp"
    android:background="@drawable/bg_home_top"
    android:clipToOutline="true"
    android:padding="16dp">

    <!-- Daily Progress Text -->
    <TextView
        android:id="@+id/tvDailyProgress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_marginTop="8dp"
        android:text="Daily Progress"
        android:textColor="#FFFFFF"
        android:textSize="16sp" />

    <!-- Notification Bell -->
    <RelativeLayout
        android:id="@+id/notificationLayout"
        android:layout_width="36dp"
        android:layout_height="36dp"
        android:layout_alignParentEnd="true"
        android:layout_marginTop="4dp">

        <ImageView
            android:id="@+id/ivBell"
            android:layout_width="36dp"
            android:layout_height="36dp"
            android:contentDescription="Notifications"
            app:srcCompat="@drawable/ic_notification_bell"
```

```
        app:tint="#FFFFFF" />

    <View
        android:id="@+id/notificationDot"
        android:layout_width="10dp"
        android:layout_height="10dp"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="4dp"
        android:layout_marginTop="4dp"
        android:background="@drawable/red_dot" />
</RelativeLayout>

<!-- Smaller Circular Progress -->
<FrameLayout
    android:layout_width="160dp"
    android:layout_height="160dp"
    android:layout_centerInParent="true">

    <ProgressBar
        android:id="@+id/circularProgressBar"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:indeterminate="false"
        android:max="100"
        android:progress="50"
        android:progressDrawable="@drawable/circular_progress_custom"
        android:rotation="-90" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/tvPercentage"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="50%"
            android:textColor="#FFFFFF"
            android:textSize="22sp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/tvDoses"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="1 of 2 doses"
            android:textColor="#FFFFFF"
            android:textSize="13sp" />
        </LinearLayout>
    </FrameLayout>
</RelativeLayout>

<!-- ✅ Quick Actions Section -->
<include
    android:id="@+id/quickActions"
    layout="@layout/layout_quick_actions"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp" />

<!-- Extra Stats -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_marginTop="16dp"
    android:paddingStart="8dp"
    android:paddingEnd="8dp">

    <TextView
        android:id="@+id/tvSkipped"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:textColor="#333333"
        android:text="Skipped: 0"/>

    <TextView
        android:id="@+id/tvActiveMeds"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:textSize="14sp"
        android:textColor="#333333"
        android:text="Active Medications: 0"/>

    <TextView
        android:id="@+id/tvDistinctTaken"
        android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"
            android:layout_marginTop="4dp"
            android:textSize="14sp"
            android:textColor="#333333"
            android:text="Distinct Taken: 0"/>
        </LinearLayout>

        <androidx.recyclerview.widget.RecyclerView
          android:id="@+id/rvMedications"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:layout_marginTop="16dp" />

      </LinearLayout>

    </androidx.core.widget.NestedScrollView>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
      android:id="@+id/fab"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentEnd="true"
      android:layout_alignParentBottom="true"
      android:layout_marginEnd="30dp"
      android:layout_marginBottom="60dp"
      android:contentDescription="Floating Action Button"
      app:srcCompat="@android:drawable/ic_dialog_email" />

</RelativeLayout>
```

Activity_SplashScreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg_analysis_gradient"
    tools:context=".Splashscreen">

    <ImageView
      android:id="@+id/logo"
      android:layout_width="195dp"
```

```
        android:layout_height="201dp"
        android:contentDescription="App Logo"
        android:src="@mipmap/logo"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:text="Smart Remainder"
        android:textColor="#00897B"
        android:textSize="28sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/logo" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Welcome_Page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F5F5F5"
    tools:context=".WelComePage">

    <!-- Main Content Card -->
    <androidx.cardview.widget.CardView
        android:id="@+id/cardSection"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="28dp"
        app:cardElevation="16dp"
        app:cardCornerRadius="28dp"
        app:cardUseCompatPadding="true"
```

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintVertical_bias="0.55">

<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:gravity="center"
   android:orientation="vertical"
   android:paddingStart="28dp"
   android:paddingEnd="28dp"
   android:paddingTop="32dp"
   android:paddingBottom="40dp">

   <!-- Circular Logo Background -->

   <FrameLayout
      android:layout_width="140dp"
      android:layout_height="140dp"
      android:padding="22dp">

      <ImageView
         android:id="@+id/logo"
         android:layout_width="97dp"
         android:layout_height="match_parent"
         android:contentDescription="App Logo"

         android:src="@mipmap/logo" />
   </FrameLayout>

   <TextView
      android:id="@+id/title"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="16dp"
      android:text="Smart Reminder"
      android:textColor="#333333"
      android:textSize="28sp"
      android:textStyle="bold"
      android:letterSpacing="0.04"/>


   <TextView
      android:id="@+id/subtitle"
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="12dp"
            android:layout_marginBottom="36dp"
            android:lineSpacingExtra="5dp"
            android:text="Your personal medication assistant that keeps you healthy
and on time."
            android:textAlignment="center"
            android:textColor="#444444"
            android:textSize="16sp" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/btnGetStarted"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:paddingTop="14dp"
            android:paddingBottom="14dp"
            android:backgroundTint="@color/teal_700"
            android:text="Get Started"
            android:textAllCaps="false"
            android:textSize="18sp"
            android:textStyle="bold"
            android:textColor="@android:color/white"
            app:cornerRadius="20dp"/>

        <TextView
            android:id="@+id/loginText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="18dp"
            android:text="Already have an account? Login"
            android:textSize="15sp"
            android:textColor="@color/teal_700"/>
      </LinearLayout>

   </androidx.cardview.widget.CardView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

SignUp.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#F5F5F5"
tools:context=".Signukt">

<ScrollView
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   app:layout_constraintBottom_toBottomOf="parent"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toTopOf="parent">

   <androidx.constraintlayout.widget.ConstraintLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:padding="24dp">

      <ImageView
         android:id="@+id/ivLogo"
         android:layout_width="100dp"
         android:layout_height="100dp"
         android:src="@mipmap/logo"
         app:layout_constraintTop_toTopOf="parent"
         app:layout_constraintStart_toStartOf="parent"
         app:layout_constraintEnd_toEndOf="parent"
         android:layout_marginTop="24dp"/>

      <TextView
         android:id="@+id/tvAppName"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_marginTop="16dp"
         android:text="Smart Remainder"
         android:textColor="@color/black"
         android:textSize="28sp"
         android:textStyle="bold"
         app:layout_constraintEnd_toEndOf="parent"
         app:layout_constraintStart_toStartOf="parent"
         app:layout_constraintTop_toBottomOf="@id/ivLogo" />

      <TextView
         android:id="@+id/tvSubtitle"
         android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:text="Create a new account"
    android:textColor="@color/black"
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvAppName" />

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/tilFullName"
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:hint="Full Name"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tvSubtitle"
    app:startIconDrawable="@drawable/ic_person">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/etFullName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPersonName" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/tilEmail"
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:hint="Email Address"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tilFullName"
    app:startIconDrawable="@drawable/ic_email">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress" />
```

```
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/tilPhone"
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:hint="Phone Number"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tilEmail"
    app:startIconDrawable="@drawable/ic_phone">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/etPhone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="phone" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/tilPassword"
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:hint="Password"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tilPhone"
    app:passwordToggleEnabled="true"
    app:startIconDrawable="@drawable/ic_lock">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/etPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/tilConfirmPassword"
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
```

```
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:hint="Confirm Password"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tilPassword"
    app:passwordToggleEnabled="true"
    app:startIconDrawable="@drawable/ic_lock">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/etConfirmPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword" />

</com.google.android.material.textfield.TextInputLayout>

<CheckBox
    android:id="@+id/cbTerms"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="I agree to the Terms &amp;amp; Conditions"
    android:textColor="@color/black"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tilConfirmPassword" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btnSignUp"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:padding="12dp"
    android:text="Sign Up"
    android:textSize="16sp"
    app:backgroundTint="#4CAF50"
    app:cornerRadius="8dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/cbTerms" />

<LinearLayout
    android:id="@+id/llLogin"
    android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"
            android:layout_marginTop="24dp"
            android:layout_marginBottom="24dp"
            android:orientation="horizontal"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/btnSignUp">

            <TextView
               android:layout_width="wrap_content"
               android:layout_height="wrap_content"
               android:text="Already have an account? "
               android:textColor="@android:color/darker_gray" />

            <TextView
               android:id="@+id/tvLogin"
               android:layout_width="wrap_content"
               android:layout_height="wrap_content"
               android:text="Login"
               android:textColor="@color/teal_700"
               android:textStyle="bold" />
         </LinearLayout>

      </androidx.constraintlayout.widget.ConstraintLayout>
   </ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Login.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="#F5F5F5"
   tools:context=".Login">

   <ImageView
      android:id="@+id/ivLogo"
      android:layout_width="100dp"
```

```
    android:layout_height="100dp"
    android:src="@mipmap/logo"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="60dp"/>

<TextView
    android:id="@+id/tvAppName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Smart Remainder"
    android:textSize="34sp"
    android:textStyle="bold"
    android:textColor="@color/black"
    app:layout_constraintTop_toBottomOf="@id/ivLogo"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp"/>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/tilEmail"
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginEnd="32dp"
    android:layout_marginTop="60dp"
    app:layout_constraintTop_toBottomOf="@id/tvAppName"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:endIconDrawable="@drawable/ic_person"
    android:hint="Email">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/tilPassword"
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
    android:layout_width="0dp"
```

```
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginEnd="32dp"
        android:layout_marginTop="16dp"
        app:layout_constraintTop_toBottomOf="@id/tilEmail"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:startIconDrawable="@drawable/ic_lock"
        app:passwordToggleEnabled="true"
        android:hint="     Password">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/etPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textPassword" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.button.MaterialButton
        android:id="@+id/btnLogin"
        android:layout_width="0dp"
        android:layout_height="60dp"
        android:text="Login"
        android:backgroundTint="#4CAF50"
        app:cornerRadius="16dp"
        android:layout_marginTop="32dp"
        app:layout_constraintTop_toBottomOf="@id/tilPassword"
        app:layout_constraintStart_toStartOf="@id/tilPassword"
        app:layout_constraintEnd_toEndOf="@id/tilPassword"/>

    <TextView
        android:id="@+id/tvSignUp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Don't have an account? Sign up"
        android:textColor="@color/teal_700"
        android:textSize="16sp"
        android:layout_marginTop="24dp"
        app:layout_constraintTop_toBottomOf="@id/btnLogin"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Add_Medicine.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F5F5F5"
    android:fillViewport="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <!-- Header with gradient background -->
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical"
            android:background="@drawable/bg_header_gradient"
            android:paddingStart="20dp"
            android:paddingEnd="20dp"
            android:paddingTop="40dp"
            android:paddingBottom="24dp"
            android:elevation="4dp">

            <ImageView
                android:id="@+id/btnBack"
                android:layout_width="40dp"
                android:layout_height="40dp"
                android:background="@drawable/bg_circle_white"
                android:padding="8dp"
                android:src="@drawable/ic_arrow_back"
                android:contentDescription="Go back"
                app:tint="@color/green_primary" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="16dp"
                android:text="New Medication"
                android:textSize="24sp"
                android:textStyle="bold"
```

```
        android:textColor="@android:color/white" />
</LinearLayout>

<!-- Main Content Card -->
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    app:cardCornerRadius="16dp"
    app:cardElevation="4dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="20dp">

        <!-- Medicine Name Section -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Medicine Information"
            android:textSize="16sp"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            android:layout_marginBottom="12dp" />

        <EditText
            android:id="@+id/etMedicineName"
            android:layout_width="match_parent"
            android:layout_height="56dp"
            android:hint="Medicine Name"
            android:inputType="textCapWords"
            android:textSize="16sp"
            android:paddingStart="16dp"
            android:paddingEnd="16dp"
            android:background="@drawable/bg_edittext_modern"
            android:drawableStart="@drawable/ic_pill"
            android:drawablePadding="12dp"
            android:layout_marginBottom="12dp"/>

        <EditText
            android:id="@+id/etDosage"
            android:layout_width="match_parent"
            android:layout_height="56dp"
            android:hint="Dosage (e.g., 500mg)"
```

```
            android:inputType="text"
            android:textSize="16sp"
            android:paddingStart="16dp"
            android:paddingEnd="16dp"
            android:background="@drawable/bg_edittext_modern"
            android:drawableStart="@drawable/ic_dosage"
            android:drawablePadding="12dp"
            android:layout_marginBottom="24dp"/>

        <!-- Frequency Section -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="How often?"
            android:textSize="16sp"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            android:layout_marginBottom="12dp" />

        <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:columnCount="3"
            android:rowCount="2"
            android:alignmentMode="alignMargins"
            android:layout_marginBottom="24dp">

            <Button
                android:id="@+id/btnOnceDaily"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Once Daily" />

            <Button
                android:id="@+id/btnTwiceDaily"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
```

```
                android:textColor="@color/modern_button_text_selector"
                android:text="Twice Daily" />

            <Button
                android:id="@+id/btnThriceDaily"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Thrice Daily" />

            <Button
                android:id="@+id/btnEveryHour"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Every Hour" />

            <Button
                android:id="@+id/btnEvery6Hours"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Every 6h" />

            <Button
                android:id="@+id/btnEvery12Hours"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Every 12h" />
```

```
        </GridLayout>

        <!-- Duration Section -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="For how long?"
            android:textSize="16sp"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            android:layout_marginBottom="12dp" />

        <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:columnCount="3"
            android:rowCount="2"
            android:alignmentMode="alignMargins"
            android:layout_marginBottom="24dp">

            <Button
                android:id="@+id/btn7days"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="7 days" />

            <Button
                android:id="@+id/btn14days"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="14 days" />

            <Button
                android:id="@+id/btn30days"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
```

```
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="30 days" />

            <Button
                android:id="@+id/btn90days"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="90 days" />

            <Button
                android:id="@+id/btnOngoing"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Ongoing" />
        </GridLayout>

        <!-- Schedule Section -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Schedule"
            android:textSize="16sp"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            android:layout_marginBottom="12dp" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="56dp"
            android:orientation="horizontal"
            android:gravity="center_vertical"
            android:background="@drawable/bg_selector_modern"
```

```
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:layout_marginBottom="12dp"
    android:clickable="true"
    android:focusable="true">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_calendar"
        app:tint="@android:color/black" />

    <TextView
        android:id="@+id/tvDate"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginStart="12dp"
        android:text="Select Start Date"
        android:textSize="16sp"
        android:textColor="@android:color/black" />

    <ImageView
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:src="@drawable/ic_arrow_forward"
        app:tint="@android:color/darker_gray" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="56dp"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:background="@drawable/bg_selector_modern"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:layout_marginBottom="24dp"
    android:clickable="true"
    android:focusable="true">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_time"
        app:tint="@android:color/black" />
```

```
    <TextView
        android:id="@+id/tvTime"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginStart="12dp"
        android:text="Select Medication Time"
        android:textSize="16sp"
        android:textColor="@android:color/black" />

    <ImageView
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:src="@drawable/ic_arrow_forward"
        app:tint="@android:color/darker_gray" />
</LinearLayout>
<!-- Inventory Section -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Inventory"
    android:textSize="16sp"
    android:textStyle="bold"
    android:textColor="@android:color/black"
    android:layout_marginBottom="12dp" />
<EditText
    android:id="@+id/etCurrentSupply"
    android:layout_width="match_parent"
    android:layout_height="56dp"
    android:hint="Current Supply"
    android:inputType="number"
    android:textSize="16sp"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:background="@drawable/bg_edittext_modern"
    android:drawableStart="@drawable/ic_inventory"
    android:drawablePadding="12dp"
    android:layout_marginBottom="12dp"/>
<EditText
    android:id="@+id/etMaxSupply"
    android:layout_width="match_parent"
    android:layout_height="56dp"
    android:hint="Max Supply"
    android:inputType="number"
    android:textSize="16sp"
```

```
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:background="@drawable/bg_edittext_modern"
    android:drawableStart="@drawable/ic_inventory"
    android:drawablePadding="12dp"
    android:layout_marginBottom="24dp"/>
<!-- Reminders Section -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Reminders &amp;amp; Tracking"
    android:textSize="16sp"
    android:textStyle="bold"
    android:textColor="@android:color/black"
    android:layout_marginBottom="12dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:background="@drawable/bg_selector_modern"
    android:padding="16dp"
    android:layout_marginBottom="12dp">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_notification"
        app:tint="@android:color/black" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginStart="12dp"
        android:text="Medication Reminders"
        android:textSize="15sp"
        android:textColor="@android:color/black" />

    <Switch
        android:id="@+id/switchReminder"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        app:trackTint="@color/switch_track_selector"
```

```
        app:thumbTint="@color/switch_thumb_selector" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:background="@drawable/bg_selector_modern"
    android:padding="16dp"
    android:layout_marginBottom="24dp">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_refill"
        app:tint="@android:color/black" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginStart="12dp"
        android:text="Refill Notifications"
        android:textSize="15sp"
        android:textColor="@android:color/black" />

    <Switch
        android:id="@+id/switchRefill"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:trackTint="@color/switch_track_selector"
        app:thumbTint="@color/switch_thumb_selector" />
</LinearLayout>

<!-- Notes Section -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Notes &amp;amp; Instructions"
    android:textSize="16sp"
    android:textStyle="bold"
    android:textColor="@android:color/black"
    android:layout_marginBottom="12dp" />

<EditText
```

```
            android:id="@+id/etNotes"
            android:layout_width="match_parent"
            android:layout_height="120dp"
            android:hint="Add special instructions, side effects, or other notes..."
            android:gravity="top|start"
            android:inputType="textMultiLine"
            android:scrollbars="vertical"
            android:padding="16dp"
            android:background="@drawable/bg_edittext_modern"
            android:textSize="15sp" />
    </LinearLayout>
</androidx.cardview.widget.CardView>

<!-- Action Buttons -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:paddingBottom="24dp">

    <com.google.android.material.button.MaterialButton
        android:id="@+id/btnAddMedication"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:text="Add Medication"
        android:textColor="@android:color/white"
        android:textSize="16sp"
        android:textStyle="bold"
        app:backgroundTint="#00897B"
        android:layout_marginBottom="12dp" />

    <com.google.android.material.button.MaterialButton
        android:id="@+id/btnCancel"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:text="Cancel"
        android:textColor="@android:color/black"
        app:strokeColor="@android:color/darker_gray"
        app:strokeWidth="1dp"
        app:backgroundTint="@android:color/white"
        android:textSize="16sp"
        android:layout_marginBottom="8dp" />
</LinearLayout>
```

```
    </LinearLayout>
</ScrollView>
```

Activity_calander.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F5F5F5"
    android:fillViewport="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <!-- Header with gradient background -->
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical"
            android:background="@drawable/bg_header_gradient"
            android:paddingStart="20dp"
            android:paddingEnd="20dp"
            android:paddingTop="40dp"
            android:paddingBottom="24dp"
            android:elevation="4dp">

            <ImageView
                android:id="@+id/btnBack"
                android:layout_width="40dp"
                android:layout_height="40dp"
                android:background="@drawable/bg_circle_white"
                android:padding="8dp"
                android:src="@drawable/ic_arrow_back"
                android:contentDescription="Go back"
                app:tint="@color/green_primary" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
```

```
        android:layout_marginStart="16dp"
        android:text="New Medication"
        android:textSize="24sp"
        android:textStyle="bold"
        android:textColor="@android:color/white" />
</LinearLayout>

<!-- Main Content Card -->
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    app:cardCornerRadius="16dp"
    app:cardElevation="4dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="20dp">

        <!-- Medicine Name Section -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Medicine Information"
            android:textSize="16sp"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            android:layout_marginBottom="12dp" />

        <EditText
            android:id="@+id/etMedicineName"
            android:layout_width="match_parent"
            android:layout_height="56dp"
            android:hint="Medicine Name"
            android:inputType="textCapWords"
            android:textSize="16sp"
            android:paddingStart="16dp"
            android:paddingEnd="16dp"
            android:background="@drawable/bg_edittext_modern"
            android:drawableStart="@drawable/ic_pill"
            android:drawablePadding="12dp"
            android:layout_marginBottom="12dp"/>

        <EditText
```

```
            android:id="@+id/etDosage"
            android:layout_width="match_parent"
            android:layout_height="56dp"
            android:hint="Dosage (e.g., 500mg)"
            android:inputType="text"
            android:textSize="16sp"
            android:paddingStart="16dp"
            android:paddingEnd="16dp"
            android:background="@drawable/bg_edittext_modern"
            android:drawableStart="@drawable/ic_dosage"
            android:drawablePadding="12dp"
            android:layout_marginBottom="24dp"/>

        <!-- Frequency Section -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="How often?"
            android:textSize="16sp"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            android:layout_marginBottom="12dp" />

        <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:columnCount="3"
            android:rowCount="2"
            android:alignmentMode="alignMargins"
            android:layout_marginBottom="24dp">

            <Button
                android:id="@+id/btnOnceDaily"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Once Daily" />

            <Button
                android:id="@+id/btnTwiceDaily"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
```

```
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Twice Daily" />

            <Button
                android:id="@+id/btnThriceDaily"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Thrice Daily" />

            <Button
                android:id="@+id/btnEveryHour"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Every Hour" />

            <Button
                android:id="@+id/btnEvery6Hours"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Every 6h" />

            <Button
                android:id="@+id/btnEvery12Hours"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"
```

```
android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Every 12h" />
        </GridLayout>

        <!-- Duration Section -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="For how long?"
            android:textSize="16sp"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            android:layout_marginBottom="12dp" />

        <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:columnCount="3"
            android:rowCount="2"
            android:alignmentMode="alignMargins"
            android:layout_marginBottom="24dp">

            <Button
                android:id="@+id/btn7days"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="7 days" />

            <Button
                android:id="@+id/btn14days"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="14 days" />
```

```
            <Button
                android:id="@+id/btn30days"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="30 days" />

            <Button
                android:id="@+id/btn90days"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="90 days" />

            <Button
                android:id="@+id/btnOngoing"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_columnWeight="1"
                android:layout_margin="4dp"

android:background="@drawable/modern_button_background_selector"
                android:textColor="@color/modern_button_text_selector"
                android:text="Ongoing" />
        </GridLayout>

        <!-- Schedule Section -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Schedule"
            android:textSize="16sp"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            android:layout_marginBottom="12dp" />

        <LinearLayout
            android:layout_width="match_parent"
```

```
    android:layout_height="56dp"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:background="@drawable/bg_selector_modern"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:layout_marginBottom="12dp"
    android:clickable="true"
    android:focusable="true">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_calendar"
        app:tint="@android:color/black" />

    <TextView
        android:id="@+id/tvDate"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginStart="12dp"
        android:text="Select Start Date"
        android:textSize="16sp"
        android:textColor="@android:color/black" />

    <ImageView
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:src="@drawable/ic_arrow_forward"
        app:tint="@android:color/darker_gray" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="56dp"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:background="@drawable/bg_selector_modern"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:layout_marginBottom="24dp"
    android:clickable="true"
    android:focusable="true">

    <ImageView
```

```
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_time"
        app:tint="@android:color/black" />

    <TextView
        android:id="@+id/tvTime"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginStart="12dp"
        android:text="Select Medication Time"
        android:textSize="16sp"
        android:textColor="@android:color/black" />

    <ImageView
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:src="@drawable/ic_arrow_forward"
        app:tint="@android:color/darker_gray" />
</LinearLayout>
<!-- Inventory Section -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Inventory"
    android:textSize="16sp"
    android:textStyle="bold"
    android:textColor="@android:color/black"
    android:layout_marginBottom="12dp" />
<EditText
    android:id="@+id/etCurrentSupply"
    android:layout_width="match_parent"
    android:layout_height="56dp"
    android:hint="Current Supply"
    android:inputType="number"
    android:textSize="16sp"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:background="@drawable/bg_edittext_modern"
    android:drawableStart="@drawable/ic_inventory"
    android:drawablePadding="12dp"
    android:layout_marginBottom="12dp"/>
<EditText
    android:id="@+id/etMaxSupply"
    android:layout_width="match_parent"
```

```
    android:layout_height="56dp"
    android:hint="Max Supply"
    android:inputType="number"
    android:textSize="16sp"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:background="@drawable/bg_edittext_modern"
    android:drawableStart="@drawable/ic_inventory"
    android:drawablePadding="12dp"
    android:layout_marginBottom="24dp"/>
<!-- Reminders Section -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Reminders &amp;amp; Tracking"
    android:textSize="16sp"
    android:textStyle="bold"
    android:textColor="@android:color/black"
    android:layout_marginBottom="12dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:background="@drawable/bg_selector_modern"
    android:padding="16dp"
    android:layout_marginBottom="12dp">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_notification"
        app:tint="@android:color/black" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginStart="12dp"
        android:text="Medication Reminders"
        android:textSize="15sp"
        android:textColor="@android:color/black" />

    <Switch
        android:id="@+id/switchReminder"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        app:trackTint="@color/switch_track_selector"
        app:thumbTint="@color/switch_thumb_selector" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:background="@drawable/bg_selector_modern"
    android:padding="16dp"
    android:layout_marginBottom="24dp">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_refill"
        app:tint="@android:color/black" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginStart="12dp"
        android:text="Refill Notifications"
        android:textSize="15sp"
        android:textColor="@android:color/black" />

    <Switch
        android:id="@+id/switchRefill"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:trackTint="@color/switch_track_selector"
        app:thumbTint="@color/switch_thumb_selector" />
</LinearLayout>

<!-- Notes Section -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Notes &amp;amp; Instructions"
    android:textSize="16sp"
    android:textStyle="bold"
```

```
            android:textColor="@android:color/black"
            android:layout_marginBottom="12dp" />

        <EditText
            android:id="@+id/etNotes"
            android:layout_width="match_parent"
            android:layout_height="120dp"
            android:hint="Add special instructions, side effects, or other notes..."
            android:gravity="top|start"
            android:inputType="textMultiLine"
            android:scrollbars="vertical"
            android:padding="16dp"
            android:background="@drawable/bg_edittext_modern"
            android:textSize="15sp" />
    </LinearLayout>
</androidx.cardview.widget.CardView>

<!-- Action Buttons -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:paddingBottom="24dp">

    <com.google.android.material.button.MaterialButton
        android:id="@+id/btnAddMedication"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:text="Add Medication"
        android:textColor="@android:color/white"
        android:textSize="16sp"
        android:textStyle="bold"
        app:backgroundTint="#00897B"
        android:layout_marginBottom="12dp" />

    <com.google.android.material.button.MaterialButton
        android:id="@+id/btnCancel"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:text="Cancel"
        android:textColor="@android:color/black"
        app:strokeColor="@android:color/darker_gray"
        app:strokeWidth="1dp"
        app:backgroundTint="@android:color/white"
```

```
            android:textSize="16sp"
            android:layout_marginBottom="8dp" />
        </LinearLayout>
    </LinearLayout>
</ScrollView>
```

Data_analysis.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg_dark_blue_theme"
    tools:context=".data_anaysis">

    <ImageButton
        android:id="@+id/analysis_back_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:background="@drawable/bg_circle_white"
        android:contentDescription="Go back"
        android:padding="8dp"
        android:src="@drawable/ic_arrow_back"
        app:layout_constraintBottom_toBottomOf="@+id/titleTextView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/titleTextView"
        app:tint="#3949AB" />

    <TextView
        android:id="@+id/titleTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:text="Reminder Analysis"
        android:textColor="@android:color/white"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent"/>

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginTop="16dp"
    android:fillViewport="true"
    app:layout_constraintTop_toBottomOf="@id/titleTextView"
    app:layout_constraintBottom_toBottomOf="parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <!-- Weekly Adherence Card -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Weekly Adherence"
            android:textColor="@android:color/white"
            android:textSize="18sp"
            android:textStyle="bold"
            android:layout_marginBottom="8dp"/>

        <androidx.cardview.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:cardCornerRadius="8dp"
            android:layout_marginBottom="24dp">

            <com.github.mikephil.charting.charts.CombinedChart
                android:id="@+id/medicationCombinedChart"
                android:layout_width="match_parent"
                android:layout_height="250dp"
                android:padding="12dp"/>

        </androidx.cardview.widget.CardView>

        <!-- Overall Adherence Card -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Overall Adherence"
            android:textColor="@android:color/white"
```

```
    android:textSize="18sp"
    android:textStyle="bold"
    android:layout_marginBottom="8dp"/>

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:cardCornerRadius="8dp"
    android:layout_marginBottom="24dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="12dp">

        <com.github.mikephil.charting.charts.PieChart
            android:id="@+id/medicationPieChart"
            android:layout_width="match_parent"
            android:layout_height="220dp"/>

        <LinearLayout
            android:id="@+id/pieChartLegend"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center"
            android:layout_marginTop="8dp"/>
    </LinearLayout>

</androidx.cardview.widget.CardView>

<!-- Response Time Analysis Card -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Response Time Analysis"
    android:textColor="@android:color/white"
    android:textSize="18sp"
    android:textStyle="bold"
    android:layout_marginBottom="8dp"/>

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:cardCornerRadius="8dp"
```

```
    android:layout_marginBottom="24dp">

    <com.github.mikephil.charting.charts.ScatterChart
        android:id="@+id/medicationScatterChart"
        android:layout_width="match_parent"
        android:layout_height="220dp"
        android:padding="12dp"/>

</androidx.cardview.widget.CardView>

<!-- Adherence Accuracy Card -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Adherence Accuracy"
    android:textColor="@android:color/white"
    android:textSize="18sp"
    android:textStyle="bold"
    android:layout_marginBottom="8dp"/>

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:cardCornerRadius="8dp"
    android:layout_marginBottom="24dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center"
        android:padding="16dp">

        <TextView
            android:id="@+id/tvAccuracyPercentage"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="--%"
            android:textColor="@color/text_primary"
            android:textSize="60sp"
            android:textStyle="bold"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Overall Accuracy"
```

```
                android:textColor="@color/text_secondary"
                android:layout_marginTop="4dp"
                android:textSize="16sp"/>
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <!-- ML Insight Card -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ML Insight"
        android:textColor="@android:color/white"
        android:textSize="18sp"
        android:textStyle="bold"
        android:layout_marginBottom="8dp"/>

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="8dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:padding="16dp">

            <ImageView
                android:layout_width="40dp"
                android:layout_height="40dp"
                android:src="@drawable/ic_ml_brain"
                android:layout_marginEnd="16dp"
                android:layout_gravity="center_vertical"/>

            <TextView
                android:id="@+id/mlInsightText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textColor="@color/text_primary"
                android:textSize="14sp"
                android:text="This feature is ready! Add your trained
'adherence_model.tflite' to the assets folder to unlock live predictions."/>
        </LinearLayout>
    </androidx.cardview.widget.CardView>

</LinearLayout>
```

```
    </ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_history.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F5F5F5"
    android:orientation="vertical"
    tools:context=".HistoryActivity">

    <!-- Header -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:background="@drawable/bg_header_gradient"
        android:gravity="center_vertical"
        android:orientation="horizontal"
        android:paddingStart="16dp"
        android:paddingTop="40dp"
        android:paddingEnd="16dp"
        android:paddingBottom="16dp">

        <ImageButton
            android:id="@+id/backButton"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:background="@drawable/bg_circle_white"
            android:padding="8dp"
            android:src="@drawable/ic_arrow_back"
            android:contentDescription="Go back"
            app:tint="@color/green_primary"  />

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:gravity="center"
            android:orientation="horizontal">

            <TextView
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="10dp"
            android:letterSpacing="0.01"
            android:text="History Log"
            android:textColor="#FFFFFF"
            android:textSize="20sp"
            android:textStyle="bold" />
    </LinearLayout>

    <View
        android:layout_width="50dp"
        android:layout_height="40dp" />

</LinearLayout>

<!-- Filter Tabs -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF"
    android:gravity="center"
    android:orientation="horizontal"
    android:padding="16dp">

    <TextView
        android:id="@+id/btnFilterAll"
        android:layout_width="0dp"
        android:layout_height="36dp"
        android:layout_marginEnd="8dp"
        android:layout_weight="1"
        android:background="@drawable/filter_button_selected"
        android:gravity="center"
        android:text="All"
        android:textColor="#FFFFFF"
        android:textSize="14sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/btnFilterTaken"
        android:layout_width="0dp"
        android:layout_height="36dp"
        android:layout_marginEnd="8dp"
        android:layout_weight="1"
        android:background="@drawable/filter_button_unselected"
        android:gravity="center"
```

```
        android:text="Taken"
        android:textColor="#757575"
        android:textSize="14sp" />

    <TextView
        android:id="@+id/btnFilterSkipped"
        android:layout_width="0dp"
        android:layout_height="36dp"
        android:layout_weight="1"
        android:background="@drawable/filter_button_unselected"
        android:gravity="center"
        android:text="Skipped"
        android:textColor="#757575"
        android:textSize="14sp" />
</LinearLayout>

<!-- History List -->
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/historyRecyclerView"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:clipToPadding="false"
    android:padding="16dp" />

<!-- Clear All Button -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF"
    android:gravity="center"
    android:padding="20dp">

    <TextView
        android:id="@+id/btnClearAll"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/clear_button_bg"
        android:drawableStart="@android:drawable/ic_menu_delete"
        android:drawablePadding="8dp"
        android:drawableTint="#EF5350"
        android:paddingStart="24dp"
        android:paddingTop="12dp"
        android:paddingEnd="24dp"
        android:paddingBottom="12dp"
        android:text="Clear All Data"
```

```
            android:textColor="#EF5350"
            android:textSize="14sp"
            android:textStyle="bold" />
    </LinearLayout>

</LinearLayout>
```

Profile.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#FFFFFF">

    <!-- Toolbar -->
    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#4CAF50"
        android:paddingTop="24dp"
        android:paddingBottom="16dp"
        android:elevation="4dp"
        app:navigationIcon="@drawable/ic_back">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Profile"
            android:textColor="#FFFFFF"
            android:textSize="20sp"
            android:textStyle="bold"
            android:layout_gravity="center"/>
    </androidx.appcompat.widget.Toolbar>

    <!-- Profile Content -->
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="24dp"
    android:gravity="center_horizontal">

    <!-- Profile Picture with Edit Icon -->
    <FrameLayout
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:layout_marginTop="24dp">

        <androidx.cardview.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            app:cardCornerRadius="60dp"
            app:cardElevation="4dp">

            <ImageView
                android:id="@+id/ivProfilePicture"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:scaleType="centerCrop"
                android:src="@drawable/ic_user" />
        </androidx.cardview.widget.CardView>

        <ImageView
            android:id="@+id/ivEditProfileImage"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:layout_gravity="bottom|end"
            android:background="@drawable/bg_add_icon"
            android:src="@drawable/ic_add_photo"
            android:padding="10dp"
            android:elevation="6dp" />

    </FrameLayout>

    <!-- User Name -->
    <TextView
        android:id="@+id/tvUserName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="John Doe"
        android:textSize="24sp"
```

```
    android:textStyle="bold"
    android:textColor="#000000"
    android:layout_marginTop="16dp"/>

<!-- Username -->
<TextView
    android:id="@+id/tvUsername"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@john_doe"
    android:textSize="14sp"
    android:textColor="#666666"
    android:layout_marginTop="4dp"/>

<!-- Email Section -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="16dp"
    android:background="@drawable/info_background"
    android:layout_marginTop="32dp"
    android:gravity="center_vertical">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_email"
        app:tint="#666666"/>

    <TextView
        android:id="@+id/tvEmail"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="john.doe@email.com"
        android:textSize="14sp"
        android:textColor="#333333"
        android:layout_marginStart="16dp"/>
</LinearLayout>

<!-- Phone Section -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
```

```
            android:padding="16dp"
            android:background="@drawable/info_background"
            android:layout_marginTop="12dp"
            android:gravity="center_vertical">

            <ImageView
                android:layout_width="24dp"
                android:layout_height="24dp"
                android:src="@drawable/ic_phone"
                app:tint="#666666"/>

            <TextView
                android:id="@+id/tvPhone"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Phone: +1 234 567 8901"
                android:textSize="14sp"
                android:textColor="#333333"
                android:layout_marginStart="16dp"/>
        </LinearLayout>

        <!-- Edit Profile Button -->
        <com.google.android.material.button.MaterialButton
            android:id="@+id/btnEditProfile"
            android:layout_width="match_parent"
            android:layout_height="56dp"
            android:text="Edit Profile"
            app:backgroundTint="@color/blue_dark"
            android:textSize="16sp"
            android:layout_marginTop="32dp"/>

        <!-- Logout Button -->
        <com.google.android.material.button.MaterialButton
            android:id="@+id/btnLogout"
            android:layout_width="match_parent"
            android:layout_height="56dp"
            android:text="Logout"
            app:backgroundTint="@color/chart_red"
            android:textSize="16sp"
            android:layout_marginTop="16dp"/>

    </LinearLayout>
  </ScrollView>
</LinearLayout>
```

Refil.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#FFFFFF">

    <!-- Toolbar -->
    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#4CAF50"
        android:paddingTop="24dp"
        android:paddingBottom="16dp"
        android:elevation="4dp"
        app:navigationIcon="@drawable/ic_back">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Profile"
            android:textColor="#FFFFFF"
            android:textSize="20sp"
            android:textStyle="bold"
            android:layout_gravity="center"/>
    </androidx.appcompat.widget.Toolbar>

    <!-- Profile Content -->
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:padding="24dp"
            android:gravity="center_horizontal">

            <!-- Profile Picture with Edit Icon -->
            <FrameLayout
```

```
    android:layout_width="120dp"
    android:layout_height="120dp"
    android:layout_marginTop="24dp">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:cardCornerRadius="60dp"
        app:cardElevation="4dp">

        <ImageView
            android:id="@+id/ivProfilePicture"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:scaleType="centerCrop"
            android:src="@drawable/ic_user" />
    </androidx.cardview.widget.CardView>

    <ImageView
        android:id="@+id/ivEditProfileImage"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_gravity="bottom|end"
        android:background="@drawable/bg_add_icon"
        android:src="@drawable/ic_add_photo"
        android:padding="10dp"
        android:elevation="6dp" />

</FrameLayout>

<!-- User Name -->
<TextView
    android:id="@+id/tvUserName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="John Doe"
    android:textSize="24sp"
    android:textStyle="bold"
    android:textColor="#000000"
    android:layout_marginTop="16dp"/>

<!-- Username -->
<TextView
    android:id="@+id/tvUsername"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:text="@john_doe"
    android:textSize="14sp"
    android:textColor="#666666"
    android:layout_marginTop="4dp"/>

<!-- Email Section -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="16dp"
    android:background="@drawable/info_background"
    android:layout_marginTop="32dp"
    android:gravity="center_vertical">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_email"
        app:tint="#666666"/>

    <TextView
        android:id="@+id/tvEmail"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="john.doe@email.com"
        android:textSize="14sp"
        android:textColor="#333333"
        android:layout_marginStart="16dp"/>
</LinearLayout>

<!-- Phone Section -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="16dp"
    android:background="@drawable/info_background"
    android:layout_marginTop="12dp"
    android:gravity="center_vertical">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/ic_phone"
```

```
            app:tint="#666666"/>

        <TextView
            android:id="@+id/tvPhone"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Phone: +1 234 567 8901"
            android:textSize="14sp"
            android:textColor="#333333"
            android:layout_marginStart="16dp"/>
    </LinearLayout>

    <!-- Edit Profile Button -->
    <com.google.android.material.button.MaterialButton
        android:id="@+id/btnEditProfile"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:text="Edit Profile"
        app:backgroundTint="@color/blue_dark"
        android:textSize="16sp"
        android:layout_marginTop="32dp"/>

    <!-- Logout Button -->
    <com.google.android.material.button.MaterialButton
        android:id="@+id/btnLogout"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:text="Logout"
        app:backgroundTint="@color/chart_red"
        android:textSize="16sp"
        android:layout_marginTop="16dp"/>

  </LinearLayout>
 </ScrollView>
</LinearLayout>
```

Chat_Pop_Up.xml file

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/chatPopupLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
```

```
    android:padding="12dp"
    android:background="@android:color/white">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:fillViewport="true"
        tools:ignore="SpeakableTextPresentCheck">

        <LinearLayout
            android:id="@+id/chatLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical" />
    </ScrollView>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:paddingTop="8dp">

        <EditText
            android:id="@+id/etMessage"
            android:layout_width="0dp"
            android:layout_weight="1"
            android:layout_height="wrap_content"
            android:hint="Ask something..."
            android:minHeight="48dp"
            android:background="@drawable/rounded_input"
            android:padding="10dp" />

        <Button
            android:id="@+id/btnSend"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Send"
            android:textColor="@android:color/white"
            android:backgroundTint="#6200EE"
            android:layout_marginStart="8dp"/>
    </LinearLayout>
</LinearLayout>
```

Editprofile.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="24dp">

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/etName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>
    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:layout_marginTop="16dp">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/etUsername"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>
    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:layout_marginTop="16dp">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/etEmail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textEmailAddress"/>
    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone"
        android.layout_marginTop="16dp">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/etPhone"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="phone"/>
    </com.google.android.material.textfield.TextInputLayout>
</LinearLayout>
```

Item_history.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginVertical="8dp"
    app:cardCornerRadius="8dp"
    app:cardElevation="4dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="16dp">

        <View
            android:id="@+id/colorBar"
            android:layout_width="4dp"
            android:layout_height="match_parent"
            android:background="#4CAF50" />

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:orientation="vertical"
            android:layout_marginStart="16dp">

            <TextView
```

```
            android:id="@+id/medicationName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Paracetamol"
            android:textColor="#000000"
            android:textSize="16sp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/medicationDosage"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="500mg"
            android:textColor="#757575"
            android:textSize="14sp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="end">

        <TextView
            android:id="@+id/medicationTime"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="08:00"
            android:textColor="#000000"
            android:textSize="16sp" />

        <TextView
            android:id="@+id/medicationStatus"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Taken"
            android:textColor="#4CAF50"
            android:textSize="14sp" />
    </LinearLayout>
  </LinearLayout>

</androidx.cardview.widget.CardView>
```

Item_history_event.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="16dp">

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="vertical">

        <TextView
            android:id="@+id/tvMedicationName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Medication Name" />

        <TextView
            android:id="@+id/tvMedicationDose"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Dose" />

        <TextView
            android:id="@+id/tvMedicationTime"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Time" />

    </LinearLayout>

    <TextView
        android:id="@+id/tvMedicationStatus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Status" />

</LinearLayout>
```

Item_medication.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginVertical="8dp"
    app:cardCornerRadius="8dp"
    app:cardElevation="4dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="16dp">

        <View
            android:id="@+id/colorBar"
            android:layout_width="4dp"
            android:layout_height="match_parent"
            android:background="#4CAF50" />

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:orientation="vertical"
            android:layout_marginStart="16dp">

            <TextView
                android:id="@+id/medicationName"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Paracetamol"
                android:textColor="#000000"
                android:textSize="16sp"
                android:textStyle="bold" />

            <TextView
                android:id="@+id/medicationDosage"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="500mg"
                android:textColor="#757575"
```

```
            android:textSize="14sp" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:gravity="end">

            <TextView
                android:id="@+id/medicationTime"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="08:00"
                android:textColor="#000000"
                android:textSize="16sp" />

            <TextView
                android:id="@+id/medicationStatus"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Take"
                android:textColor="#4CAF50"
                android:textSize="14sp" />
        </LinearLayout>
    </LinearLayout>

</androidx.cardview.widget.CardView>
```

Item_refil_medicine.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="12dp"
    app:cardCornerRadius="12dp"
    app:cardElevation="2dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
```

```
android:padding="16dp">

<!-- Color Indicator -->
<View
    android:id="@+id/colorIndicator"
    android:layout_width="4dp"
    android:layout_height="match_parent"
    android:layout_marginEnd="12dp"
    android:background="#FFA726" />

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:orientation="vertical">

    <!-- Medicine Name and Status -->
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/tvMedicineName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Paracetamol"
            android:textColor="#000000"
            android:textSize="16sp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/tvSupplyStatus"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:text="Good"
            android:textColor="#4CAF50"
            android:textSize="12sp"
            android:textStyle="bold" />

    </RelativeLayout>

    <!-- Dosage -->
    <TextView
        android:id="@+id/tvDosage"
        android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_marginTop="2dp"
    android:text="500mg"
    android:textColor="#666666"
    android:textSize="12sp" />

<!-- Progress Bar and Supply -->
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp">

    <ProgressBar
        android:id="@+id/progressBar"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="8dp"
        android:layout_centerVertical="true"
        android:max="100"
        android:progress="50"
        android:progressBackgroundTint="#E0E0E0"
        android:progressTint="#4CAF50" />

    <TextView
        android:id="@+id/tvCurrentSupply"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:text="0 units"
        android:textColor="#000000"
        android:textSize="12sp"
        android:textStyle="bold" />

</RelativeLayout>

<!-- Refill Button -->
<Button
    android:id="@+id/btnRecordRefill"
    style="@style/Widget.AppCompat.Button.Borderless"
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:layout_gravity="end"
    android:layout_marginTop="12dp"
    android:background="@drawable/button_background"
    android:paddingStart="16dp"
```

```
                android:paddingEnd="16dp"
                android:text="Record Refill"
                android:textAllCaps="false"
                android:textColor="#999999" />

        </LinearLayout>
    </LinearLayout>
</androidx.cardview.widget.CardView>
```

Layout_progress_section.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="220dp"
    android:background="@drawable/bg_home_top" android:padding="16dp"
    android:clipToOutline="true">

    <TextView
        android:id="@+id/tvDailyProgress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Daily Progress"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        android:textStyle="bold"
        android:layout_alignParentStart="true"
        android:layout_marginTop="8dp" />

    <ImageView
        android:id="@+id/ivBell"
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_alignParentEnd="true"
        android:layout_marginTop="8dp"
        app:srcCompat="@drawable/ic_notification_bell"
        android:contentDescription="Notifications"
        app:tint="#FFFFFF" />

    <View
        android:id="@+id/notificationDot"
        android:layout_width="8dp"
```

```
    android:layout_height="8dp"
    android:layout_alignTop="@id/ivBell"
    android:layout_alignEnd="@id/ivBell"
    android:layout_marginEnd="2dp"
    android:background="@drawable/red_dot" />

<FrameLayout
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/tvDailyProgress"
    android:layout_marginTop="-20dp">

    <ProgressBar
        android:id="@+id/circularProgressBar"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:indeterminate="false"
        android:max="100"
        android:progress="50"
        android:progressDrawable="@drawable/circular_progress"
        android:rotation="-90"/>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="100dp"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/tvPercentage"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="50%"
            android:textSize="36sp"
            android:textColor="#FFFFFF"
            android:textStyle="bold"/>

        <TextView
            android:id="@+id/tvDoses"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="1 of 2 doses"
```

```
            android:textSize="14sp"
            android:textColor="#FFFFFF"/>
        </LinearLayout>
    </FrameLayout>

</RelativeLayout>
```

Layout_quick_actions.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/quickActionsSection"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:columnCount="2"
    android:rowCount="3"
    android:padding="8dp"
    android:layout_marginBottom="12dp">

    <!-- Add Medication -->
    <LinearLayout
        android:id="@+id/llAddMedication"
        android:layout_width="0dp"
        android:layout_height="120dp"
        android:layout_columnWeight="1"
        android:layout_rowWeight="1"
        android:orientation="vertical"
        android:gravity="center"
        android:layout_margin="8dp"
        android:background="@drawable/bg_add_medication"
        android:padding="12dp"
        android:clickable="true"
        android:focusable="true">

        <ImageView
            android:layout_width="32dp"
            android:layout_height="32dp"
            android:src="@android:drawable/ic_input_add"
            app:tint="@android:color/white" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Add Medication"
```

```
        android:textColor="@android:color/white"
        android:textStyle="bold"
        android:textSize="16sp"
        android:layout_marginTop="8dp" />
</LinearLayout>

<!-- Calendar View -->
<LinearLayout
    android:id="@+id/llCalendarView"
    android:layout_width="0dp"
    android:layout_height="120dp"
    android:layout_columnWeight="1"
    android:layout_rowWeight="1"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_margin="8dp"
    android:background="@drawable/bg_calendar"
    android:padding="12dp"
    android:clickable="true"
    android:focusable="true">

    <ImageView
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:src="@android:drawable/ic_menu_month"
        app:tint="@android:color/white" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Calendar View"
        android:textColor="@android:color/white"
        android:textStyle="bold"
        android:textSize="16sp"
        android:layout_marginTop="8dp" />
</LinearLayout>

<!-- History Log -->
<LinearLayout
    android:id="@+id/llHistoryLog"
    android:layout_width="0dp"
    android:layout_height="120dp"
    android:layout_columnWeight="1"
    android:layout_rowWeight="1"
    android:orientation="vertical"
    android:gravity="center"
```

```
    android:layout_margin="8dp"
    android:background="@drawable/bg_history"
    android:padding="12dp"
    android:clickable="true"
    android:focusable="true">

    <ImageView
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:src="@android:drawable/ic_menu_recent_history"
        app:tint="@android:color/white" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="History Log"
        android:textColor="@android:color/white"
        android:textStyle="bold"
        android:textSize="16sp"
        android:layout_marginTop="8dp" />
</LinearLayout>

<!-- Refill Tracker -->
<LinearLayout
    android:id="@+id/llRefillTracker"
    android:layout_width="0dp"
    android:layout_height="120dp"
    android:layout_columnWeight="1"
    android:layout_rowWeight="1"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_margin="8dp"
    android:background="@drawable/bg_refill"
    android:padding="12dp"
    android:clickable="true"
    android:focusable="true">

    <ImageView
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:src="@android:drawable/ic_menu_manage"
        app:tint="@android:color/white" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:text="Refill Tracker"
        android:textColor="@android:color/white"
        android:textStyle="bold"
        android:textSize="16sp"
        android:layout_marginTop="8dp"/>
</LinearLayout>

<!-- 🧍 User Profile -->
<LinearLayout
    android:id="@+id/llUserProfile"
    android:layout_width="0dp"
    android:layout_height="120dp"
    android:layout_columnWeight="1"
    android:layout_rowWeight="1"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_margin="8dp"
    android:background="@drawable/bg_profile"
    android:padding="12dp"
    android:clickable="true"
    android:focusable="true">

    <ImageView
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:src="@drawable/ic_chart"
        app:tint="@android:color/white" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="View Analytics"
        android:textColor="@android:color/white"
        android:textStyle="bold"
        android:textSize="16sp"
        android:layout_marginTop="8dp"/>
</LinearLayout>

<!-- ☎️ Emergency Call -->
<LinearLayout
    android:id="@+id/llEmergencyCall"
    android:layout_width="0dp"
    android:layout_height="120dp"
    android:layout_columnWeight="1"
    android:layout_rowWeight="1"
```

```
        android:orientation="vertical"
        android:gravity="center"
        android:layout_margin="8dp"
        android:background="@drawable/bg_emergency_call"
        android:padding="12dp"
        android:clickable="true"
        android:focusable="true">

        <ImageView
            android:layout_width="32dp"
            android:layout_height="32dp"
            android:src="@android:drawable/ic_menu_call"
            app:tint="@android:color/white" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Emergency Call"
            android:textColor="@android:color/white"
            android:textStyle="bold"
            android:textSize="16sp"
            android:layout_marginTop="8dp"/>
    </LinearLayout>

</GridLayout>
```

5. Kt code

AddMedicineActivity.kt file

```
package com.example.smartremainder

import android.app.DatePickerDialog
import android.app.TimePickerDialog
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.smartremainder.databinding.ActivityAddMedicineBinding
import java.util.Calendar

class AddMedicineActivity : AppCompatActivity() {

    private lateinit var binding: ActivityAddMedicineBinding
```

```
    private lateinit var dbHelper: DatabaseHelper
    private lateinit var alarmScheduler: AlarmScheduler

    private var selectedFrequency: String? = null
    private var selectedDuration: String? = null
    private var userEmail: String? = null

    private lateinit var frequencyButtons: List<Button>
    private lateinit var durationButtons: List<Button>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityAddMedicineBinding.inflate(layoutInflater)
        setContentView(binding.root)

        userEmail = intent.getStringExtra("USER_EMAIL")
        if (userEmail == null) {
            Toast.makeText(this, "Error: User not identified.",
Toast.LENGTH_LONG).show()
            finish()
            return
        }

        dbHelper = DatabaseHelper(this)
        alarmScheduler = AlarmScheduler(this)

        initializeButtonGroups()
        setupClickListeners()
    }

    private fun initializeButtonGroups() {
        frequencyButtons = listOf(
            binding.btnOnceDaily, binding.btnTwiceDaily, binding.btnThriceDaily,
            binding.btnEveryHour, binding.btnEvery6Hours, binding.btnEvery12Hours
        )
        durationButtons = listOf(
            binding.btn7days, binding.btn14days, binding.btn30days,
            binding.btn90days, binding.btnOngoing
        )
    }

    private fun setupClickListeners() {
        binding.btnBack.setOnClickListener { finish() }
        binding.btnCancel.setOnClickListener { finish() }

        binding.tvDate.setOnClickListener { showDatePickerDialog() }
```

```
    binding.tvTime.setOnClickListener { showTimePickerDialog() }

    frequencyButtons.forEach { button ->
        button.setOnClickListener {
            handleSelection(button, frequencyButtons)
            selectedFrequency = button.text.toString()
        }
    }

    durationButtons.forEach { button ->
        button.setOnClickListener {
            handleSelection(button, durationButtons)
            selectedDuration = button.text.toString()
        }
    }

    binding.btnAddMedication.setOnClickListener {
        saveMedication()
    }
}

private fun handleSelection(selectedButton: Button, group: List<Button>) {
    group.forEach { button ->
        button.isSelected = (button.id == selectedButton.id)
    }
}

private fun showDatePickerDialog() {
    val calendar = Calendar.getInstance()
    val datePickerDialog = DatePickerDialog(
        this,
        { _, year, month, dayOfMonth ->
            val date = "$year-${month + 1}-$dayOfMonth"
            binding.tvDate.text = date
        },
        calendar.get(Calendar.YEAR),
        calendar.get(Calendar.MONTH),
        calendar.get(Calendar.DAY_OF_MONTH)
    )
    datePickerDialog.show()
}

private fun showTimePickerDialog() {
    val calendar = Calendar.getInstance()
    val timePickerDialog = TimePickerDialog(
        this,
```

```
            { _, hourOfDay, minute ->
                val time = String.format("%02d:%02d", hourOfDay, minute)
                binding.tvTime.text = time
            },
            calendar.get(Calendar.HOUR_OF_DAY),
            calendar.get(Calendar.MINUTE),
            true
        )
        timePickerDialog.show()
    }

    private fun parseDuration(durationString: String): Int {
        if (durationString.equals("Ongoing", ignoreCase = true)) {
            return -1 // Use -1 to represent an ongoing duration
        }
        // Extracts digits from strings like "7 days"
        return durationString.filter { it.isDigit() }.toIntOrNull() ?: 0
    }

    private fun saveMedication() {
        val medicineName = binding.etMedicineName.text.toString().trim()
        val dosage = binding.etDosage.text.toString().trim()
        val startDate = binding.tvDate.text.toString()
        val time = binding.tvTime.text.toString()
        val notes = binding.etNotes.text.toString().trim()
        val reminderEnabled = binding.switchReminder.isChecked
        val refillEnabled = binding.switchRefill.isChecked
        val currentSupply = binding.etCurrentSupply.text.toString().toIntOrNull() ?: 0
        val maxSupply = binding.etMaxSupply.text.toString().toIntOrNull() ?: 0

        if (medicineName.isBlank() || selectedFrequency == null || selectedDuration ==
null || startDate == "Select Start Date" || time == "Select Medication Time") {
            Toast.makeText(this, "Please fill all required fields.",
Toast.LENGTH_SHORT).show()
            return
        }

        val durationInDays = parseDuration(selectedDuration!!)

        val newMedication = Medication(
            id = 0, // ID is auto-generated by the database
            userEmail = userEmail!!,
            name = medicineName,
            dosage = dosage,
            frequency = selectedFrequency!!,
            duration = durationInDays,
```

```
            startDate = startDate,
            times = listOf(time), // The data model expects a list of times
            reminderEnabled = reminderEnabled,
            refillEnabled = refillEnabled,
            notes = notes,
            image = null, // No image selection is implemented in this screen
            currentSupply = currentSupply,
            maxSupply = maxSupply
        )

        val id = dbHelper.addMedication(newMedication)

        if (id != -1L) {
            if (reminderEnabled) {
                // The alarm scheduler needs the medication object with the newly assigned
ID
                val scheduledMedication = newMedication.copy(id = id)
                alarmScheduler.schedule(scheduledMedication)
            }
            Toast.makeText(this, "Medication added successfully!",
Toast.LENGTH_SHORT).show()
            finish()
        } else {
            Toast.makeText(this, "Failed to add medication.",
Toast.LENGTH_SHORT).show()
        }
    }
}
```

AlarmReceiver.kt file

```
package com.example.smartremainder

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch

class AlarmReceiver : BroadcastReceiver() {

    override fun onReceive(context: Context, intent: Intent) {
        val medicationId = intent.getLongExtra("MEDICATION_ID", -1L)
```

```
        val userEmail = intent.getStringExtra("USER_EMAIL")

        if (medicationId == -1L || userEmail == null) return
        // Start the ringtone service
        val ringtoneIntent = Intent(context, RingtoneService::class.java)
        context.startService(ringtoneIntent)

        val pendingResult = goAsync()

        CoroutineScope(Dispatchers.IO).launch {
            try {
                val dbHelper = DatabaseHelper(context)
                val notificationHelper = NotificationHelper(context)
                val alarmScheduler = AlarmScheduler(context)

                // Get medication details
                val medication = dbHelper.getMedicationById(medicationId, userEmail)

                medication?.let { med ->

                    // STEP 1 — INSERT DOSE LOG (Increase Total Dose)
                    dbHelper.insertDoseLog(
                        medicationId = med.id,
                        userEmail = userEmail,
                        status = "Pending"
                    )

                    // STEP 2 — Show Notification
                    notificationHelper.showNotification(med)

                    // STEP 3 — Reschedule next alarm
                    alarmScheduler.schedule(med)

                    // STEP 4 — Notify Activity/Fragment to refresh UI
                    context.sendBroadcast(
                        Intent("com.example.smartremainder.MEDICATION_UPDATE")
                    )
                }

            } finally {
                pendingResult.finish()
            }
        }
    }
}
```

AlarmSchedular.kt file

```
package com.example.smartremainder

import android.app.AlarmManager
import android.app.PendingIntent
import android.content.Context
import android.content.Intent
import android.os.Build
import java.util.Calendar

class AlarmScheduler(private val context: Context) {

    private val alarmManager = context.getSystemService(Context.ALARM_SERVICE)
as AlarmManager

    fun schedule(medication: Medication) {
        medication.times.forEachIndexed { index, time ->
            val timeParts = time.split(":")
            if (timeParts.size != 2) return@forEachIndexed // or log an error

            val hour = timeParts[0].toIntOrNull() ?: return@forEachIndexed
            val minute = timeParts[1].toIntOrNull() ?: return@forEachIndexed

            val calendar = Calendar.getInstance().apply {
                timeInMillis = System.currentTimeMillis()
                set(Calendar.HOUR_OF_DAY, hour)
                set(Calendar.MINUTE, minute)
                set(Calendar.SECOND, 0)
                set(Calendar.MILLISECOND, 0)

                // If the time has already passed for today, schedule it for tomorrow
                if (before(Calendar.getInstance())) {
                    add(Calendar.DATE, 1)
                }
            }

            // Create a unique request code for each time to ensure PendingIntents are not
overwritten
            // Using medication id and time index to create a unique code.
            val requestCode = medication.id.toInt() * 100 + index

            val intent = Intent(context, AlarmReceiver::class.java).apply {
                putExtra("MEDICATION_ID", medication.id)
                putExtra("USER_EMAIL", medication.userEmail)
            }
```

```
            val pendingIntent = PendingIntent.getBroadcast(
                context,
                requestCode,
                intent,
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                    PendingIntent.FLAG_UPDATE_CURRENT or
PendingIntent.FLAG_IMMUTABLE
                } else {
                    PendingIntent.FLAG_UPDATE_CURRENT
                }
            )

            // This schedules a one-time exact alarm. The app will need to reschedule the
next one
            // after this alarm fires, probably within the AlarmReceiver.
            alarmManager.setExactAndAllowWhileIdle(
                AlarmManager.RTC_WAKEUP,
                calendar.timeInMillis,
                pendingIntent
            )
        }
    }

    fun cancel(medication: Medication) {
        medication.times.forEachIndexed { index, _ ->
            // Create a unique request code for each time to ensure we cancel the correct
PendingIntent
            val requestCode = medication.id.toInt() * 100 + index

            val intent = Intent(context, AlarmReceiver::class.java).apply {
                putExtra("MEDICATION_ID", medication.id)
                putExtra("USER_EMAIL", medication.userEmail)
            }

            val pendingIntent = PendingIntent.getBroadcast(
                context,
                requestCode,
                intent,
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                    PendingIntent.FLAG_NO_CREATE or
PendingIntent.FLAG_IMMUTABLE
                } else {
                    PendingIntent.FLAG_NO_CREATE
                }
            )
```

```
            pendingIntent?.let {
                alarmManager.cancel(it)
                it.cancel() // Also cancel the pending intent itself
            }
        }
    }
}
```

BootReceiver.kt file

```
package com.example.smartremainder

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.util.Log

class BootReceiver : BroadcastReceiver() {

    override fun onReceive(context: Context, intent: Intent) {
        if (intent.action == "android.intent.action.BOOT_COMPLETED") {
            Log.d("BootReceiver", "Boot completed, rescheduling alarms...")
            val dbHelper = DatabaseHelper(context)
            val alarmScheduler = AlarmScheduler(context)

            // Get all medications for all users
            val allMedications = dbHelper.getAllMedicationsForAllUsers()

            if (allMedications.isEmpty()) {
                Log.d("BootReceiver", "No medications found to reschedule.")
                return
            }

            // Loop through each medication and reschedule the alarm
            for (medication in allMedications) {
                if (medication.reminderEnabled) {
                    Log.d("BootReceiver", "Rescheduling alarm for med ID:
${medication.id} for user: ${medication.userEmail}")
                    alarmScheduler.schedule(medication)
                } else {
                    Log.d("BootReceiver", "Skipping med ID: ${medication.id} for user:
${medication.userEmail} (reminders disabled)")
```

```
            }
        }
        Log.d("BootReceiver", "Finished rescheduling ${allMedications.size}
alarms.")
    }
  }
}
```

CalanderActivity.kt file

```
package com.example.smartremainder

import android.os.Bundle
import android.widget.ImageButton
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.GridLayoutManager
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import android.widget.TextView
import java.text.SimpleDateFormat
import java.util.*

class CalendarActivity : AppCompatActivity() {

    private lateinit var calendarRecyclerView: RecyclerView
    private lateinit var medicationsRecyclerView: RecyclerView
    private lateinit var tvMonthYear: TextView
    private lateinit var tvSelectedDate: TextView
    private lateinit var btnPrevMonth: ImageButton
    private lateinit var btnNextMonth: ImageButton
    private lateinit var backButton: ImageButton

    private lateinit var calendarAdapter: CalendarAdapter
    private lateinit var medicationAdapter: MedicationAdapter

    private val calendar = Calendar.getInstance()
    private var selectedDate = Calendar.getInstance()

    private lateinit var dbHelper: DatabaseHelper
    private var allMedications: List<Medication> = emptyList()
    private var userEmail: String = ""

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

```
    dbHelper = DatabaseHelper(this)
    userEmail = intent.getStringExtra("USER_EMAIL") ?: ""

    initViews()
    loadAllMedications() // Load all medications from DB
    setupCalendar()
    setupMedicationList()
    updateMonthYear()
    updateSelectedDate() // Set initial selected date text
    updateMedicationList() // Show medications for the current date initially
}

private fun initViews() {
    calendarRecyclerView = findViewById(R.id.calendarRecyclerView)
    medicationsRecyclerView = findViewById(R.id.medicationsRecyclerView)
    tvMonthYear = findViewById(R.id.tvMonthYear)
    tvSelectedDate = findViewById(R.id.tvSelectedDate)
    btnPrevMonth = findViewById(R.id.btnPrevMonth)
    btnNextMonth = findViewById(R.id.btnNextMonth)
    backButton = findViewById(R.id.calendar_back_button) // Use the unique ID

    backButton.setOnClickListener {
        finish() // Go back to the previous screen
    }

    btnPrevMonth.setOnClickListener {
        calendar.add(Calendar.MONTH, -1)
        updateMonthYear()
        updateCalendar()
    }

    btnNextMonth.setOnClickListener {
        calendar.add(Calendar.MONTH, 1)
        updateMonthYear()
        updateCalendar()
    }
}

private fun loadAllMedications() {
    allMedications = dbHelper.getAllMedications(userEmail)
}

private fun setupCalendar() {
    calendarRecyclerView.layoutManager = GridLayoutManager(this, 7)
```

```
        calendarAdapter = CalendarAdapter { day ->
            if (day.isNotEmpty()) {
                val dayNum = day.toIntOrNull()
                if (dayNum != null) {
                    selectedDate.set(Calendar.YEAR, calendar.get(Calendar.YEAR))
                    selectedDate.set(Calendar.MONTH, calendar.get(Calendar.MONTH))
                    selectedDate.set(Calendar.DAY_OF_MONTH, dayNum)
                    updateSelectedDate()
                    updateMedicationList()
                }
            }
        }

        calendarRecyclerView.adapter = calendarAdapter
        updateCalendar()
    }

    private fun setupMedicationList() {
        medicationsRecyclerView.layoutManager = LinearLayoutManager(this)
        medicationAdapter = MedicationAdapter()
        medicationsRecyclerView.adapter = medicationAdapter
    }

    private fun updateCalendar() {
        val days = mutableListOf<String>()
        val tempCalendar = calendar.clone() as Calendar

        tempCalendar.set(Calendar.DAY_OF_MONTH, 1)
        val firstDayOfMonth = tempCalendar.get(Calendar.DAY_OF_WEEK) - 1 //
Sunday is 1, etc.

        // Add empty slots for days before the month starts
        repeat(firstDayOfMonth) {
            days.add("")
        }

        // Add days of the month
        val maxDay = tempCalendar.getActualMaximum(Calendar.DAY_OF_MONTH)
        for (day in 1..maxDay) {
            days.add(day.toString())
        }

        calendarAdapter.updateDays(days,
selectedDate.get(Calendar.DAY_OF_MONTH))
    }
```

```
    private fun updateMonthYear() {
        val format = SimpleDateFormat("MMMM yyyy", Locale.getDefault())
        tvMonthYear.text = format.format(calendar.time)
    }

    private fun updateSelectedDate() {
        val format = SimpleDateFormat("EEEE, dd MMMM", Locale.getDefault())
        tvSelectedDate.text = format.format(selectedDate.time)
    }

    private fun updateMedicationList() {
        val dateFormat = SimpleDateFormat("yyyy-MM-dd", Locale.getDefault())
        val selectedDateString = dateFormat.format(selectedDate.time)

        val medicationsForDate = allMedications.filter {
            // This logic assumes `startDate` is in "yyyy-MM-dd" format.
            it.startDate == selectedDateString
        }

        medicationAdapter.updateMedications(medicationsForDate)
    }
}
```

CalanderAdapter.kt file

```
package com.example.smartremainder

import android.graphics.Color
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class CalendarAdapter(private val onDayClick: (String) -> Unit) :
RecyclerView.Adapter<CalendarAdapter.CalendarViewHolder>() {

    private val days = mutableListOf<String>()
    private var selectedDay = -1

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
CalendarViewHolder {
        val view =
```

```
LayoutInflater.from(parent.context).inflate(R.layout.item_calendar_day, parent, false)
        return CalendarViewHolder(view)
    }

    override fun onBindViewHolder(holder: CalendarViewHolder, position: Int) {
        val day = days[position]
        holder.dayText.text = day

        if (day.isNotEmpty() && day.toInt() == selectedDay) {
            holder.dayText.setBackgroundColor(Color.CYAN)
        } else {
            holder.dayText.setBackgroundColor(Color.TRANSPARENT)
        }

        holder.itemView.setOnClickListener {
            onDayClick(day)
        }
    }

    override fun getItemCount() = days.size

    fun updateDays(newDays: List<String>, selected: Int) {
        days.clear()
        days.addAll(newDays)
        selectedDay = selected
        notifyDataSetChanged()
    }

    class CalendarViewHolder(itemView: View) :
RecyclerView.ViewHolder(itemView) {
        val dayText: TextView = itemView.findViewById(R.id.dayText)
    }
}
```

ChatBotFragment.kt file

```
package com.example.smartremainder

import android.os.Bundle
import android.view.Gravity
import android.view.LayoutInflater
import android.view.View
```

```
import android.view.ViewGroup
import android.widget.Button
import android.widget.EditText
import android.widget.LinearLayout
import android.widget.TextView
import androidx.fragment.app.DialogFragment
import com.example.smartremainder.R

class ChatbotFragment : DialogFragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.chat_popup, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        val etMessage = view.findViewById<EditText>(R.id.etMessage)
        val btnSend = view.findViewById<Button>(R.id.btnSend)
        val chatLayout = view.findViewById<LinearLayout>(R.id.chatLayout)

        // Initial bot greeting
        val initBotMsg = TextView(requireContext())
        initBotMsg.text = "Hi 👋 How can I assist you today?"
        initBotMsg.setBackgroundResource(R.drawable.bot_bubble)
        initBotMsg.setPadding(20, 10, 20, 10)
        val initParams = LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.WRAP_CONTENT,
            LinearLayout.LayoutParams.WRAP_CONTENT
        )
        initParams.setMargins(8, 8, 8, 8)
        initParams.gravity = Gravity.START
        initBotMsg.layoutParams = initParams
        chatLayout.addView(initBotMsg)

        btnSend.setOnClickListener {
            val userMsg = etMessage.text.toString().trim()
            if (userMsg.isNotEmpty()) {
                val userTextView = TextView(requireContext())
                userTextView.text = userMsg
                userTextView.setBackgroundResource(R.drawable.user_bubble)
                userTextView.setPadding(20, 10, 20, 10)
                val userParams = LinearLayout.LayoutParams(
```

```
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT
            )
            userParams.setMargins(8, 8, 8, 8)
            userParams.gravity = Gravity.END
            userTextView.layoutParams = userParams
            chatLayout.addView(userTextView)
            etMessage.setText("")

            // Gemini API call
            GeminiService.sendMessage(userMsg) { reply ->
                activity?.runOnUiThread {
                    val botTextView = TextView(requireContext())
                    botTextView.text = reply
                    botTextView.setBackgroundResource(R.drawable.bot_bubble)
                    botTextView.setPadding(20, 10, 20, 10)
                    val botParams = LinearLayout.LayoutParams(
                        LinearLayout.LayoutParams.WRAP_CONTENT,
                        LinearLayout.LayoutParams.WRAP_CONTENT
                    )
                    botParams.setMargins(8, 8, 8, 8)
                    botParams.gravity = Gravity.START
                    botTextView.layoutParams = botParams
                    chatLayout.addView(botTextView)
                }
            }
        }
    }

    override fun onStart() {
        super.onStart()
        dialog?.window?.setLayout(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.WRAP_CONTENT)
    }
}
```

DataAnalysis.kt file

```
package com.example.smartremainder

import android.graphics.Color
import android.graphics.Typeface
```

```
import android.os.Bundle
import android.util.Log
import android.view.Gravity
import android.widget.ImageButton
import android.widget.ImageView
import android.widget.LinearLayout
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.example.smartremainder.databinding.ActivityDataAnaysisBinding
import com.github.mikephil.charting.charts.CombinedChart
import com.github.mikephil.charting.charts.ScatterChart
import com.github.mikephil.charting.components.XAxis
import com.github.mikephil.charting.data.*
import com.github.mikephil.charting.formatter.IndexAxisValueFormatter
import com.github.mikephil.charting.formatter.ValueFormatter
import java.text.DecimalFormat

class data_anaysis : AppCompatActivity() {

    private lateinit var binding: ActivityDataAnaysisBinding
    private lateinit var dbHelper: DatabaseHelper
    private lateinit var userEmail: String

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityDataAnaysisBinding.inflate(layoutInflater)
        setContentView(binding.root)

        dbHelper = DatabaseHelper(this)
        userEmail = intent.getStringExtra("USER_EMAIL") ?: ""

        val backButton = findViewById<ImageButton>(R.id.analysis_back_button)
        backButton.setOnClickListener {
            finish()
        }

        loadAnalyticsData()
    }

    private fun loadAnalyticsData() {
        try {
            val weeklyAdherence = dbHelper.getWeeklyAdherenceStats(userEmail)
            val overallAdherence = dbHelper.getOverallAdherenceStats(userEmail)
            val responseTimeData = dbHelper.getResponseTimeStats(userEmail)

            // Debug logs
```

```
            Log.d("DataAnalysis", "Weekly Adherence size: ${weeklyAdherence?.size}")
            Log.d("DataAnalysis", "Overall Adherence: $overallAdherence")
            Log.d("DataAnalysis", "Response Time size: ${responseTimeData?.size}")

            // Setup charts with null checks
            if (weeklyAdherence.isNotEmpty()) {
                setupCombinedChart(weeklyAdherence)
            } else {
                Log.w("DataAnalysis", "No weekly adherence data")
            }

            setupPieChart(overallAdherence)
            setupAccuracyCard(overallAdherence)

            if (responseTimeData.isNotEmpty()) {
                setupScatterChart(responseTimeData)
            } else {
                Log.w("DataAnalysis", "No response time data")
            }

            generateSmartInsight()
        } catch (e: Exception) {
            Log.e("DataAnalysis", "Error loading analytics: ${e.message}", e)
        }
    }

    private fun generateSmartInsight() {
        try {
            val worstDay = dbHelper.getLowestAdherenceDay(userEmail)
            val insightText = if (worstDay.isNotBlank()) {
                "Our analysis suggests you struggle most on ${worstDay}s. Try setting an
extra reminder on those days!"
            } else {
                "You are on a great track! No specific problem days found. Keep up the
consistent work."
            }
            binding.mlInsightText.text = insightText
        } catch (e: Exception) {
            Log.e("DataAnalysis", "Error generating insight: ${e.message}", e)
            binding.mlInsightText.text = "Keep tracking your medications to get
personalized insights!"
        }
    }

    private fun setupCombinedChart(weeklyAdherence: List<BarEntry>) {
        try {
```

```
        val combinedChart = binding.medicationCombinedChart
        combinedChart.clear()

        combinedChart.description.isEnabled = false
        combinedChart.drawOrder = arrayOf(CombinedChart.DrawOrder.BAR,
CombinedChart.DrawOrder.LINE)
        combinedChart.setBackgroundColor(Color.WHITE)
        combinedChart.setDrawGridBackground(false)

        // Bar dataset with lighter color
        val barDataSet = BarDataSet(weeklyAdherence, "Weekly Adherence").apply {
            color = Color.parseColor("#E3F2FD")
            valueTextColor = Color.BLACK
            valueTextSize = 10f
            setDrawValues(true)
            valueFormatter = object : ValueFormatter() {
                override fun getFormattedValue(value: Float): String {
                    return "${value.toInt()}%"
                }
            }
        }
        val barData = BarData(barDataSet)
        barData.barWidth = 0.8f

        // Line dataset with smooth curve
        val lineEntries = weeklyAdherence.map { Entry(it.x, it.y) }
        val lineDataSet = LineDataSet(lineEntries, "Adherence Trend").apply {
            color = Color.parseColor("#5C6BC0")
            lineWidth = 3f
            setCircleColor(Color.parseColor("#5C6BC0"))
            circleRadius = 5f
            circleHoleRadius = 2.5f
            setDrawCircleHole(true)
            setDrawValues(false)
            mode = LineDataSet.Mode.CUBIC_BEZIER
            cubicIntensity = 0.2f
        }
        val lineData = LineData(lineDataSet)

        val combinedData = CombinedData()
        combinedData.setData(barData)
        combinedData.setData(lineData)
        combinedChart.data = combinedData

        val days = arrayOf("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")
        combinedChart.xAxis.apply {
```

```
            valueFormatter = IndexAxisValueFormatter(days)
            position = XAxis.XAxisPosition.BOTTOM
            granularity = 1f
            textColor = Color.BLACK
            textSize = 11f
            setDrawGridLines(false)
            setDrawAxisLine(true)
            axisMinimum = -0.5f
            axisMaximum = 6.5f
        }

        combinedChart.axisLeft.apply {
            textColor = Color.BLACK
            textSize = 11f
            axisMinimum = 0f
            axisMaximum = 100f
            valueFormatter = object : ValueFormatter() {
                override fun getFormattedValue(value: Float): String {
                    return "${value.toInt()}%"
                }
            }
            setLabelCount(6, true)
            setDrawGridLines(true)
            gridColor = Color.LTGRAY
            gridLineWidth = 0.5f
        }

        combinedChart.axisRight.isEnabled = false

        combinedChart.legend.apply {
            textColor = Color.BLACK
            textSize = 12f
            verticalAlignment =
com.github.mikephil.charting.components.Legend.LegendVerticalAlignment.BOTTO
M
            horizontalAlignment =
com.github.mikephil.charting.components.Legend.LegendHorizontalAlignment.CEN
TER
            orientation =
com.github.mikephil.charting.components.Legend.LegendOrientation.HORIZONTAL
            setDrawInside(false)
            yOffset = 10f
        }

        combinedChart.animateY(1000)
        combinedChart.invalidate()
```

```
            Log.d("DataAnalysis", "Combined chart setup complete")
        } catch (e: Exception) {
            Log.e("DataAnalysis", "Error setting up combined chart: ${e.message}", e)
        }
    }

    private fun setupPieChart(overallAdherence: Pair<Float, Float>) {
        try {
            val pieChart = binding.medicationPieChart
            pieChart.clear()

            val takenPercent = overallAdherence.first
            val missedPercent = 100f - takenPercent

            val entries = ArrayList<PieEntry>().apply {
                if (takenPercent > 0) add(PieEntry(takenPercent, "Taken"))
                if (missedPercent > 0) add(PieEntry(missedPercent, "Missed"))
            }

            // Handle case when there's no data
            if (entries.isEmpty()) {
                entries.add(PieEntry(100f, "No Data"))
            }

            val colors = if (entries.size == 1 && entries[0].label == "No Data") {
                listOf(Color.LTGRAY)
            } else {
                listOf(
                    Color.parseColor("#66BB6A"), // Green
                    Color.parseColor("#EF5350")  // Red
                )
            }

            val dataSet = PieDataSet(entries, "").apply {
                this.colors = colors
                setDrawValues(false)
                sliceSpace = 2f
            }

            pieChart.apply {
                data = PieData(dataSet)
                description.isEnabled = false
                isDrawHoleEnabled = true
                holeRadius = 70f
                transparentCircleRadius = 75f
```

```
            setHoleColor(Color.WHITE)
            legend.isEnabled = false
            centerText = "Overall"
            setCenterTextSize(22f)
            setCenterTextColor(Color.GRAY)
            setCenterTextTypeface(Typeface.DEFAULT_BOLD)
            isRotationEnabled = true
            isHighlightPerTapEnabled = true
            animateY(1000)
            invalidate()
        }

        createPieChartLegend()
        Log.d("DataAnalysis", "Pie chart setup complete: Taken=${takenPercent}%")
    } catch (e: Exception) {
        Log.e("DataAnalysis", "Error setting up pie chart: ${e.message}", e)
    }
}

private fun createPieChartLegend() {
    binding.pieChartLegend.removeAllViews()
    val legendData = listOf(
        Pair("Taken", Color.parseColor("#66BB6A")),
        Pair("Missed", Color.parseColor("#EF5350"))
    )

    for (item in legendData) {
        val legendItem = LinearLayout(this).apply {
            orientation = LinearLayout.HORIZONTAL
            gravity = Gravity.CENTER_VERTICAL
            layoutParams = LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT
            ).apply {
                marginEnd = 32
            }
        }

        val colorBox = ImageView(this).apply {
            layoutParams = LinearLayout.LayoutParams(30, 30)
            setBackgroundColor(item.second)
        }

        val label = TextView(this).apply {
            text = item.first
            textSize = 16f
```

```
            setTextColor(Color.BLACK)
            layoutParams = LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT
            ).apply {
                marginStart = 12
            }
        }

        legendItem.addView(colorBox)
        legendItem.addView(label)
        binding.pieChartLegend.addView(legendItem)
    }
}

private fun setupScatterChart(responseTimeData: List<Entry>) {
    try {
        val scatterChart = binding.medicationScatterChart
        scatterChart.clear()

        val dataSet = ScatterDataSet(responseTimeData, "Response Time").apply {
            setColor(Color.parseColor("#FFA726")) // Orange
            setScatterShape(ScatterChart.ScatterShape.CIRCLE)
            scatterShapeSize = 20f
            setDrawValues(false)
        }

        scatterChart.apply {
            data = ScatterData(dataSet)
            description.isEnabled = false
            legend.textColor = Color.BLACK
            legend.textSize = 12f

            xAxis.apply {
                position = XAxis.XAxisPosition.BOTTOM
                textColor = Color.BLACK
                textSize = 11f
                axisMinimum = 0f
                axisMaximum = 24f
                granularity = 4f
                valueFormatter = object : ValueFormatter() {
                    override fun getFormattedValue(value: Float): String {
                        return "${value.toInt()}h"
                    }
                }
                setDrawGridLines(true)
```

```
                gridColor = Color.LTGRAY
            }

            axisLeft.apply {
                textColor = Color.BLACK
                textSize = 11f
                axisMinimum = 0f
                axisMaximum = 60f
                valueFormatter = object : ValueFormatter() {
                    override fun getFormattedValue(value: Float): String {
                        return "${value.toInt()}m"
                    }
                }
                setLabelCount(7, true)
                setDrawGridLines(true)
                gridColor = Color.LTGRAY
            }

            axisRight.isEnabled = false
            animateXY(1000, 1000)
            invalidate()
        }

        Log.d("DataAnalysis", "Scatter chart setup complete with
${responseTimeData.size} points")
    } catch (e: Exception) {
        Log.e("DataAnalysis", "Error setting up scatter chart: ${e.message}", e)
    }
}

private fun setupAccuracyCard(overallAdherence: Pair<Float, Float>) {
    try {
        val accuracyPercent = overallAdherence.first
        val formattedPercent = if (accuracyPercent == 0f) {
            "0.0"
        } else {
            DecimalFormat("#.0").format(accuracyPercent)
        }
        binding.tvAccuracyPercentage.text = "$formattedPercent%"
        Log.d("DataAnalysis", "Accuracy card: $formattedPercent%")
    } catch (e: Exception) {
        Log.e("DataAnalysis", "Error setting up accuracy card: ${e.message}", e)
        binding.tvAccuracyPercentage.text = "0.0%"
    }
}
}
```

DataBaseHelper.kt file

```
package com.example.smartremainder

import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
import com.github.mikephil.charting.data.BarEntry
import com.github.mikephil.charting.data.Entry
import java.text.SimpleDateFormat
import java.util.*

class DatabaseHelper(context: Context) : SQLiteOpenHelper(context,
DATABASE_NAME, null, DATABASE_VERSION) {

    companion object {
        private const val DATABASE_NAME = "smart_remainder.db"
        private const val DATABASE_VERSION = 2

        // User Table
        private const val TABLE_USERS = "users"
        private const val COLUMN_USER_ID = "id"
        private const val COLUMN_USER_NAME = "name"
        private const val COLUMN_USER_EMAIL = "email"
        private const val COLUMN_USER_PASSWORD = "password"
        private const val COLUMN_USER_PHONE = "phone"
        private const val COLUMN_USER_PROFILE_PIC = "profile_pic"

        // Medication Table
        private const val TABLE_MEDICATIONS = "medications"
        private const val COLUMN_MED_ID = "id"
        private const val COLUMN_MED_USER_EMAIL = "user_email"
        private const val COLUMN_MED_NAME = "name"
        private const val COLUMN_MED_DOSAGE = "dosage"
        private const val COLUMN_MED_FREQUENCY = "frequency"
        private const val COLUMN_MED_TIMES = "times"
        private const val COLUMN_MED_DURATION = "duration"
        private const val COLUMN_MED_START_DATE = "start_date"
        private const val COLUMN_MED_REMINDER_ENABLED =
```

```
"reminder_enabled"
        private const val COLUMN_MED_REFILL_ENABLED = "refill_enabled"
        private const val COLUMN_MED_NOTES = "notes"
        private const val COLUMN_MED_CURRENT_SUPPLY = "current_supply"
        private const val COLUMN_MED_MAX_SUPPLY = "max_supply"
        private const val COLUMN_MED_IMAGE = "image"

        // Dose Log Table
        private const val TABLE_DOSE_LOGS = "dose_logs"
        private const val COLUMN_LOG_ID = "id"
        private const val COLUMN_LOG_MED_ID = "medication_id"
        private const val COLUMN_LOG_USER_EMAIL = "user_email"
        private const val COLUMN_LOG_TIMESTAMP = "timestamp"
        private const val COLUMN_LOG_STATUS = "status" // e.g., "Taken",
"Skipped", "Pending", "Refill"
        private const val COLUMN_LOG_REFILL_AMOUNT = "refill_amount" // For
refill events
    }

    override fun onCreate(db: SQLiteDatabase) {
        val createUserTable = "CREATE TABLE $TABLE_USERS (" +
                "$COLUMN_USER_ID INTEGER PRIMARY KEY
AUTOINCREMENT," +
                "$COLUMN_USER_NAME TEXT," +
                "$COLUMN_USER_EMAIL TEXT UNIQUE," +
                "$COLUMN_USER_PASSWORD TEXT," +
                "$COLUMN_USER_PHONE TEXT," +
                "$COLUMN_USER_PROFILE_PIC TEXT)"
        db.execSQL(createUserTable)

        val createMedicationTable = "CREATE TABLE $TABLE_MEDICATIONS (" +
                "$COLUMN_MED_ID INTEGER PRIMARY KEY AUTOINCREMENT,"
+
                "$COLUMN_MED_USER_EMAIL TEXT," +
                "$COLUMN_MED_NAME TEXT," +
                "$COLUMN_MED_DOSAGE TEXT," +
                "$COLUMN_MED_FREQUENCY TEXT," +
                "$COLUMN_MED_TIMES TEXT," +
                "$COLUMN_MED_DURATION INTEGER," +
                "$COLUMN_MED_START_DATE TEXT," +
                "$COLUMN_MED_REMINDER_ENABLED INTEGER," +
                "$COLUMN_MED_REFILL_ENABLED INTEGER," +
                "$COLUMN_MED_NOTES TEXT," +
                "$COLUMN_MED_CURRENT_SUPPLY INTEGER," +
                "$COLUMN_MED_MAX_SUPPLY INTEGER," +
                "$COLUMN_MED_IMAGE TEXT," +
```

```
            "FOREIGN KEY($COLUMN_MED_USER_EMAIL) REFERENCES
$TABLE_USERS($COLUMN_USER_EMAIL))"
        db.execSQL(createMedicationTable)

        val createDoseLogTable = "CREATE TABLE $TABLE_DOSE_LOGS (" +
            "$COLUMN_LOG_ID INTEGER PRIMARY KEY AUTOINCREMENT,"
+
            "$COLUMN_LOG_MED_ID INTEGER," +
            "$COLUMN_LOG_USER_EMAIL TEXT," +
            "$COLUMN_LOG_TIMESTAMP INTEGER," +
            "$COLUMN_LOG_STATUS TEXT," +
            "$COLUMN_LOG_REFILL_AMOUNT INTEGER," +
            "FOREIGN KEY($COLUMN_LOG_MED_ID) REFERENCES
$TABLE_MEDICATIONS($COLUMN_MED_ID)," +
            "FOREIGN KEY($COLUMN_LOG_USER_EMAIL) REFERENCES
$TABLE_USERS($COLUMN_USER_EMAIL))"
        db.execSQL(createDoseLogTable)
    }

    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {
        if (oldVersion < 2) {
            // Add all new columns that were added in version 2
            db.execSQL("ALTER TABLE $TABLE_MEDICATIONS ADD COLUMN
$COLUMN_MED_FREQUENCY TEXT")
            db.execSQL("ALTER TABLE $TABLE_MEDICATIONS ADD COLUMN
$COLUMN_MED_TIMES TEXT")
            db.execSQL("ALTER TABLE $TABLE_MEDICATIONS ADD COLUMN
$COLUMN_MED_DURATION INTEGER")
            db.execSQL("ALTER TABLE $TABLE_MEDICATIONS ADD COLUMN
$COLUMN_MED_START_DATE TEXT")
            db.execSQL("ALTER TABLE $TABLE_MEDICATIONS ADD COLUMN
$COLUMN_MED_REFILL_ENABLED INTEGER")
            db.execSQL("ALTER TABLE $TABLE_MEDICATIONS ADD COLUMN
$COLUMN_MED_CURRENT_SUPPLY INTEGER")
            db.execSQL("ALTER TABLE $TABLE_MEDICATIONS ADD COLUMN
$COLUMN_MED_MAX_SUPPLY INTEGER")

            // Recreate Dose Log Table with new schema
            db.execSQL("DROP TABLE IF EXISTS $TABLE_DOSE_LOGS")
            val createDoseLogTable = "CREATE TABLE $TABLE_DOSE_LOGS (" +
                "$COLUMN_LOG_ID INTEGER PRIMARY KEY
AUTOINCREMENT," +
                "$COLUMN_LOG_MED_ID INTEGER," +
                "$COLUMN_LOG_USER_EMAIL TEXT," +
                "$COLUMN_LOG_TIMESTAMP INTEGER," +
                "$COLUMN_LOG_STATUS TEXT," +
```

```
                "$COLUMN_LOG_REFILL_AMOUNT INTEGER," +
                "FOREIGN KEY($COLUMN_LOG_MED_ID) REFERENCES
$TABLE_MEDICATIONS($COLUMN_MED_ID)," +
                "FOREIGN KEY($COLUMN_LOG_USER_EMAIL) REFERENCES
$TABLE_USERS($COLUMN_USER_EMAIL))"
        db.execSQL(createDoseLogTable)
    }
}


// User Functions

fun addUser(user: User): Long {
    val db = this.writableDatabase
    val values = ContentValues()
    values.put(COLUMN_USER_NAME, user.name)
    values.put(COLUMN_USER_EMAIL, user.email)
    values.put(COLUMN_USER_PASSWORD, user.password)
    values.put(COLUMN_USER_PHONE, user.phone)
    values.put(COLUMN_USER_PROFILE_PIC, user.profilePic)
    val id = db.insert(TABLE_USERS, null, values)
    return id
}

fun getUser(email: String): User? {
    val db = this.readableDatabase
    val cursor: Cursor = db.query(
        TABLE_USERS, null,
        "$COLUMN_USER_EMAIL = ?", arrayOf(email), null, null, null
    )
    var user: User? = null
    if (cursor.moveToFirst()) {
        user = User(
            id =
cursor.getLong(cursor.getColumnIndexOrThrow(COLUMN_USER_ID)),
            name =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_USER_NAME)),
            email =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_USER_EMAIL)),
            password =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_USER_PASSWORD)),
            phone =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_USER_PHONE)),
            profilePic =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_USER_PROFILE_PIC)
)
```

```
            )
        }
        cursor.close()
        return user
    }

    fun updateUser(user: User) {
        val db = this.writableDatabase
        val values = ContentValues()
        values.put(COLUMN_USER_NAME, user.name)
        values.put(COLUMN_USER_PHONE, user.phone)
        values.put(COLUMN_USER_PROFILE_PIC, user.profilePic)

        db.update(TABLE_USERS, values, "$COLUMN_USER_EMAIL = ?",
arrayOf(user.email))
    }

    fun checkUser(email: String, passwordHash: String): Boolean {
        val db = this.readableDatabase
        val cursor = db.query(
            TABLE_USERS, arrayOf(COLUMN_USER_ID),
            "$COLUMN_USER_EMAIL = ? AND $COLUMN_USER_PASSWORD =
?",
            arrayOf(email, passwordHash),
            null, null, null
        )
        val count = cursor.count
        cursor.close()
        return count > 0
    }

    // Medication Functions
    fun addMedication(med: Medication): Long {
        val db = this.writableDatabase
        val values = ContentValues()
        values.put(COLUMN_MED_USER_EMAIL, med.userEmail)
        values.put(COLUMN_MED_NAME, med.name)
        values.put(COLUMN_MED_DOSAGE, med.dosage)
        values.put(COLUMN_MED_FREQUENCY, med.frequency)
        values.put(COLUMN_MED_TIMES, med.times.joinToString(","))
        values.put(COLUMN_MED_DURATION, med.duration)
        values.put(COLUMN_MED_START_DATE, med.startDate)
        values.put(COLUMN_MED_REMINDER_ENABLED, if
(med.reminderEnabled) 1 else 0)
        values.put(COLUMN_MED_REFILL_ENABLED, if (med.refillEnabled) 1 else
0)
```

```
        values.put(COLUMN_MED_NOTES, med.notes)
        values.put(COLUMN_MED_CURRENT_SUPPLY, med.currentSupply)
        values.put(COLUMN_MED_MAX_SUPPLY, med.maxSupply)
        values.put(COLUMN_MED_IMAGE, med.image)
        val id = db.insert(TABLE_MEDICATIONS, null, values)
        return id
    }

    fun updateMedication(med: Medication) {
        val db = this.writableDatabase
        val values = ContentValues()
        values.put(COLUMN_MED_NAME, med.name)
        values.put(COLUMN_MED_DOSAGE, med.dosage)
        values.put(COLUMN_MED_FREQUENCY, med.frequency)
        values.put(COLUMN_MED_TIMES, med.times.joinToString(","))
        values.put(COLUMN_MED_DURATION, med.duration)
        values.put(COLUMN_MED_START_DATE, med.startDate)
        values.put(COLUMN_MED_REMINDER_ENABLED, if
(med.reminderEnabled) 1 else 0)
        values.put(COLUMN_MED_REFILL_ENABLED, if (med.refillEnabled) 1 else
0)
        values.put(COLUMN_MED_NOTES, med.notes)
        values.put(COLUMN_MED_CURRENT_SUPPLY, med.currentSupply)
        values.put(COLUMN_MED_MAX_SUPPLY, med.maxSupply)
        values.put(COLUMN_MED_IMAGE, med.image)

        db.update(TABLE_MEDICATIONS, values, "$COLUMN_MED_ID = ? AND
$COLUMN_MED_USER_EMAIL = ?", arrayOf(med.id.toString(), med.userEmail))
    }

    fun deleteMedication(medicationId: Long, userEmail: String) {
        val db = this.writableDatabase
        db.delete(
            TABLE_MEDICATIONS,
            "$COLUMN_MED_ID = ? AND $COLUMN_MED_USER_EMAIL = ?",
            arrayOf(medicationId.toString(), userEmail)
        )
        // Also delete associated logs
        db.delete(
            TABLE_DOSE_LOGS,
            "$COLUMN_LOG_MED_ID = ? AND $COLUMN_LOG_USER_EMAIL =
?",
            arrayOf(medicationId.toString(), userEmail)
        )
    }
```

```
    fun getAllMedications(userEmail: String): List<Medication> {
        val medList = mutableListOf<Medication>()
        val db = this.readableDatabase
        val cursor = db.query(TABLE_MEDICATIONS, null,
"$COLUMN_MED_USER_EMAIL = ?", arrayOf(userEmail), null, null, null)

        if (cursor.moveToFirst()) {
            do {
                val med = Medication(
                    id =
cursor.getLong(cursor.getColumnIndexOrThrow(COLUMN_MED_ID)),
                    userEmail =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_USER_EMAIL))
,
                    name =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_NAME)),
                    dosage =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_DOSAGE)),
                    frequency =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_FREQUENCY))
,
                    duration =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_DURATION)),
                    startDate =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_START_DATE))
,
                    reminderEnabled =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_REMINDER_ENA
BLED)) == 1,
                    refillEnabled =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_REFILL_ENABLE
D)) == 1,
                    notes =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_NOTES)),
                    currentSupply =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_CURRENT_SUPPL
Y)),
                    maxSupply =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_MAX_SUPPLY)),
                    image =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_IMAGE)),
                    times =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_TIMES)).split(",
").map { it.trim() }
                )
                medList.add(med)
```

```
        } while (cursor.moveToNext())
    }
    cursor.close()
    return medList
}

fun getAllMedicationsForAllUsers(): List<Medication> {
    val medList = mutableListOf<Medication>()
    val db = this.readableDatabase
    val cursor = db.query(TABLE_MEDICATIONS, null, null, null, null, null, null)

    if (cursor.moveToFirst()) {
        do {
            val med = Medication(
                id =
cursor.getLong(cursor.getColumnIndexOrThrow(COLUMN_MED_ID)),
                userEmail =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_USER_EMAIL))
,
                name =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_NAME)),
                dosage =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_DOSAGE)),
                frequency =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_FREQUENCY))
,
                duration =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_DURATION)),
                startDate =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_START_DATE))
,
                reminderEnabled =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_REMINDER_ENA
BLED)) == 1,
                refillEnabled =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_REFILL_ENABLE
D)) == 1,
                notes =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_NOTES)),
                currentSupply =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_CURRENT_SUPPL
Y)),
                maxSupply =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_MAX_SUPPLY)),
                image =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_IMAGE)),
```

```
                times =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_TIMES)).split(",
").map { it.trim() }
            )
            medList.add(med)
        } while (cursor.moveToNext())
    }
    cursor.close()
    return medList
  }

  fun getMedicationById(medicationId: Long, userEmail: String): Medication? {
    val db = this.readableDatabase
    val cursor = db.query(
      TABLE_MEDICATIONS, null,
      "$COLUMN_MED_ID = ? AND $COLUMN_MED_USER_EMAIL = ?",
      arrayOf(medicationId.toString(), userEmail),
      null, null, null
    )
    var medication: Medication? = null
    if (cursor.moveToFirst()) {
      medication = Medication(
        id =
cursor.getLong(cursor.getColumnIndexOrThrow(COLUMN_MED_ID)),
        userEmail =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_USER_EMAIL))
,
        name =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_NAME)),
        dosage =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_DOSAGE)),
        frequency =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_FREQUENCY))
,
        duration =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_DURATION)),
        startDate =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_START_DATE))
,
        reminderEnabled =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_REMINDER_ENA
BLED)) == 1,
        refillEnabled =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_REFILL_ENABLE
D)) == 1,
        notes =
```

```
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_NOTES)),
            currentSupply =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_CURRENT_SUPPL
Y)),
            maxSupply =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_MED_MAX_SUPPLY)),
            image =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_IMAGE)),
            times =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_TIMES)).split(",
").map { it.trim() }
        )
      }
      cursor.close()
      return medication
   }

   fun getMedicationsForToday(userEmail: String): List<Medication> {
      val medications = getAllMedications(userEmail)
      val today = Calendar.getInstance()
      val sdf = SimpleDateFormat("yyyy-MM-dd", Locale.getDefault())

      return medications.filter { med ->
         val startDate = sdf.parse(med.startDate)
         val cal = Calendar.getInstance()
         cal.time = startDate
         val endDate = cal.apply { add(Calendar.DAY_OF_YEAR, med.duration)
}.time
         today.time.after(startDate) && today.time.before(endDate)
      }
   }


   // History / Dose Log Functions

   fun insertDoseLog(medicationId: Long, userEmail: String, status: String) {
      val db = this.writableDatabase
      val values = ContentValues()
      values.put(COLUMN_LOG_MED_ID, medicationId)
      values.put(COLUMN_LOG_USER_EMAIL, userEmail)
      values.put(COLUMN_LOG_TIMESTAMP, System.currentTimeMillis())
      values.put(COLUMN_LOG_STATUS, status)
      db.insert(TABLE_DOSE_LOGS, null, values)
   }

   fun updateMostRecentDoseStatus(medicationId: Long, userEmail: String,
```

```
newStatus: String): Boolean {
        val db = this.writableDatabase
        var updatedRows = 0
        // Find the most recent "Pending" dose for this medication
        val cursor = db.query(
            TABLE_DOSE_LOGS,
            arrayOf(COLUMN_LOG_ID),
            "$COLUMN_LOG_MED_ID = ? AND $COLUMN_LOG_USER_EMAIL =
? AND $COLUMN_LOG_STATUS = ?",
            arrayOf(medicationId.toString(), userEmail, "Pending"),
            null,
            null,
            "$COLUMN_LOG_TIMESTAMP DESC",
            "1"
        )

        if (cursor.moveToFirst()) {
            val logId =
cursor.getLong(cursor.getColumnIndexOrThrow(COLUMN_LOG_ID))
            val values = ContentValues()
            values.put(COLUMN_LOG_STATUS, newStatus)
            values.put(COLUMN_LOG_TIMESTAMP, System.currentTimeMillis()) //
Update timestamp to reflect action time

            updatedRows = db.update(TABLE_DOSE_LOGS, values,
"$COLUMN_LOG_ID = ?", arrayOf(logId.toString()))

            // If the dose was taken, decrement the supply
            if (newStatus == "Taken" && updatedRows > 0) {
                db.execSQL("UPDATE $TABLE_MEDICATIONS SET
$COLUMN_MED_CURRENT_SUPPLY =
$COLUMN_MED_CURRENT_SUPPLY - 1 WHERE $COLUMN_MED_ID = ?",
arrayOf(medicationId.toString()))
            }
        }
        cursor.close()
        return updatedRows > 0
    }


    fun markMostRecentDoseAsTaken(medicationId: Long, userEmail: String):
Boolean {
        return updateMostRecentDoseStatus(medicationId, userEmail, "Taken")
    }

    fun markMostRecentDoseAsSkipped(medicationId: Long, userEmail: String):
```

```
Boolean {
        return updateMostRecentDoseStatus(medicationId, userEmail, "Skipped")
    }


    fun getAllHistory(userEmail: String): List<MedicationHistory> {
        return getFilteredHistory(userEmail, "All")
    }

    fun getFilteredHistory(userEmail: String, filter: String): List<MedicationHistory> {
        val historyList = mutableListOf<MedicationHistory>()
        val db = this.readableDatabase

        val query = "SELECT l.$COLUMN_LOG_ID,
l.$COLUMN_LOG_TIMESTAMP, l.$COLUMN_LOG_STATUS,
m.$COLUMN_MED_NAME, m.$COLUMN_MED_DOSAGE " +
                "FROM $TABLE_DOSE_LOGS l JOIN $TABLE_MEDICATIONS m ON
l.$COLUMN_LOG_MED_ID = m.$COLUMN_MED_ID " +
                "WHERE l.$COLUMN_LOG_USER_EMAIL = ? " +
                (if (filter != "All") "AND l.$COLUMN_LOG_STATUS = ? " else "") +
                "ORDER BY l.$COLUMN_LOG_TIMESTAMP DESC"

        val selectionArgs = if (filter != "All") arrayOf(userEmail, filter) else
arrayOf(userEmail)

        val cursor: Cursor = db.rawQuery(query, selectionArgs)

        if (cursor.moveToFirst()) {
            do {
                historyList.add(
                    MedicationHistory(
                        logId =
cursor.getLong(cursor.getColumnIndexOrThrow(COLUMN_LOG_ID)),
                        timestamp =
cursor.getLong(cursor.getColumnIndexOrThrow(COLUMN_LOG_TIMESTAMP)),
                        status =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_LOG_STATUS)),
                        dosage =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_DOSAGE)),
                        medicationName =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_MED_NAME))
                    )
                )
            } while (cursor.moveToNext())
        }
        cursor.close()
```

```
        return historyList
    }

    fun clearAllHistory(userEmail: String) {
        val db = this.writableDatabase
        db.delete(TABLE_DOSE_LOGS, "$COLUMN_LOG_USER_EMAIL = ?",
arrayOf(userEmail))
    }


    // Progress and Stats Functions
    fun getTodayTakenDoses(userEmail: String): Int {
        val db = this.readableDatabase
        val todayStart = getStartOfToday()
        val cursor = db.rawQuery(
            "SELECT COUNT(*) FROM $TABLE_DOSE_LOGS WHERE
$COLUMN_LOG_USER_EMAIL = ? AND $COLUMN_LOG_STATUS = 'Taken'
AND $COLUMN_LOG_TIMESTAMP >= ?",
            arrayOf(userEmail, todayStart.toString())
        )
        var count = 0
        if (cursor.moveToFirst()) {
            count = cursor.getInt(0)
        }
        cursor.close()
        return count
    }

    fun getTodayTotalDoses(userEmail: String): Int {
        val medications = getMedicationsForToday(userEmail)
        return medications.sumOf { it.times.size }
    }


    fun recordRefill(medicationId: Long, userEmail: String, refillAmount: Int) {
        val db = this.writableDatabase
        // Update medication supply
        db.execSQL(
            "UPDATE $TABLE_MEDICATIONS SET
$COLUMN_MED_CURRENT_SUPPLY = $COLUMN_MED_MAX_SUPPLY
WHERE $COLUMN_MED_ID = ?",
            arrayOf(medicationId.toString())
        )
        // Log the refill event
        val values = ContentValues()
        values.put(COLUMN_LOG_MED_ID, medicationId)
```

```
        values.put(COLUMN_LOG_USER_EMAIL, userEmail)
        values.put(COLUMN_LOG_TIMESTAMP, System.currentTimeMillis())
        values.put(COLUMN_LOG_STATUS, "Refill")
        values.put(COLUMN_LOG_REFILL_AMOUNT, refillAmount)
        db.insert(TABLE_DOSE_LOGS, null, values)
    }


    fun getWeeklyAdherenceStats(userEmail: String): List<BarEntry> {
        val db = this.readableDatabase
        val entries = mutableListOf<BarEntry>()
        val calendar = Calendar.getInstance()

        for (i in 6 downTo 0) {
            calendar.time = Date()
            calendar.add(Calendar.DAY_OF_YEAR, -i)
            val dayStart = getStartOfDay(calendar).timeInMillis
            val dayEnd = getEndOfDay(calendar).timeInMillis

            val totalCursor = db.rawQuery("SELECT COUNT(*) FROM
$TABLE_DOSE_LOGS WHERE $COLUMN_LOG_USER_EMAIL = ? AND
$COLUMN_LOG_TIMESTAMP BETWEEN ? AND ?", arrayOf(userEmail,
dayStart.toString(), dayEnd.toString()))
            var total = 0
            if (totalCursor.moveToFirst()) total = totalCursor.getInt(0)
            totalCursor.close()

            val takenCursor = db.rawQuery("SELECT COUNT(*) FROM
$TABLE_DOSE_LOGS WHERE $COLUMN_LOG_USER_EMAIL = ? AND
$COLUMN_LOG_STATUS = 'Taken' AND $COLUMN_LOG_TIMESTAMP
BETWEEN ? AND ?", arrayOf(userEmail, dayStart.toString(), dayEnd.toString()))
            var taken = 0
            if (takenCursor.moveToFirst()) taken = takenCursor.getInt(0)
            takenCursor.close()

            val adherence = if (total > 0) (taken.toFloat() / total.toFloat()) * 100f else 0f
            entries.add(BarEntry(6f - i, adherence))
        }
        return entries
    }

    fun getOverallAdherenceStats(userEmail: String): Pair<Float, Float> {
        val db = this.readableDatabase
        val totalCursor = db.rawQuery("SELECT COUNT(*) FROM
$TABLE_DOSE_LOGS WHERE $COLUMN_LOG_USER_EMAIL = ? AND
($COLUMN_LOG_STATUS = 'Taken' OR $COLUMN_LOG_STATUS =
```

```
'Skipped')", arrayOf(userEmail))
        var total = 0
        if (totalCursor.moveToFirst()) total = totalCursor.getInt(0)
        totalCursor.close()

        val takenCursor = db.rawQuery("SELECT COUNT(*) FROM
$TABLE_DOSE_LOGS WHERE $COLUMN_LOG_USER_EMAIL = ? AND
$COLUMN_LOG_STATUS = 'Taken'", arrayOf(userEmail))
        var taken = 0
        if (takenCursor.moveToFirst()) taken = takenCursor.getInt(0)
        takenCursor.close()


        if (total == 0) return Pair(0f, 0f)

        val takenPercent = (taken.toFloat() / total.toFloat()) * 100f
        val missedPercent = 100f - takenPercent
        return Pair(takenPercent, missedPercent)
    }

    fun getResponseTimeStats(userEmail: String): List<Entry> {
        // This would require storing scheduled time vs. action time.
        // For now, returning an empty list to avoid a crash.
        return emptyList()
    }

    fun getLowestAdherenceDay(userEmail: String): String {
        val db = this.readableDatabase
        // More complex query needed here, for now, return a default
        return "Not enough data"
    }

    // Helper functions for dates
    private fun getStartOfToday(): Long {
        val calendar = Calendar.getInstance()
        calendar.set(Calendar.HOUR_OF_DAY, 0)
        calendar.set(Calendar.MINUTE, 0)
        calendar.set(Calendar.SECOND, 0)
        calendar.set(Calendar.MILLISECOND, 0)
        return calendar.timeInMillis
    }
    private fun getStartOfDay(calendar: Calendar): Calendar {
        calendar.set(Calendar.HOUR_OF_DAY, 0)
        calendar.set(Calendar.MINUTE, 0)
        calendar.set(Calendar.SECOND, 0)
        calendar.set(Calendar.MILLISECOND, 0)
```

```
        return calendar
    }

    private fun getEndOfDay(calendar: Calendar): Calendar {
        calendar.set(Calendar.HOUR_OF_DAY, 23)
        calendar.set(Calendar.MINUTE, 59)
        calendar.set(Calendar.SECOND, 59)
        calendar.set(Calendar.MILLISECOND, 999)
        return calendar
    }
}
```

HistoryActivity.kt file

```
package com.example.smartremainder

import android.os.Bundle
import android.widget.ImageButton
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

class HistoryActivity : AppCompatActivity() {

    private lateinit var dbHelper: DatabaseHelper
    private lateinit var historyRecyclerView: RecyclerView
    private lateinit var historyAdapter: HistoryAdapter
    private var userEmail: String = ""

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

        userEmail = intent.getStringExtra("USER_EMAIL") ?: ""

        dbHelper = DatabaseHelper(this)
        historyRecyclerView = findViewById(R.id.historyRecyclerView)
        historyRecyclerView.layoutManager = LinearLayoutManager(this)

        // Initialize adapter with an empty list
        historyAdapter = HistoryAdapter(emptyList())
```

```
        historyRecyclerView.adapter = historyAdapter

        val backButton: ImageButton = findViewById(R.id.backButton)
        backButton.setOnClickListener {
            onBackPressedDispatcher.onBackPressed()
        }

        val btnFilterAll: TextView = findViewById(R.id.btnFilterAll)
        val btnFilterTaken: TextView = findViewById(R.id.btnFilterTaken)
        val btnFilterSkipped: TextView = findViewById(R.id.btnFilterSkipped)
        val btnClearAll: TextView = findViewById(R.id.btnClearAll)

        loadHistory("all") // Initial load

        btnFilterAll.setOnClickListener {
            loadHistory("all")
            updateFilterButtons(btnFilterAll, btnFilterTaken, btnFilterSkipped)
        }

        btnFilterTaken.setOnClickListener {
            loadHistory("taken")
            updateFilterButtons(btnFilterTaken, btnFilterAll, btnFilterSkipped)
        }

        btnFilterSkipped.setOnClickListener {
            loadHistory("skipped")
            updateFilterButtons(btnFilterSkipped, btnFilterAll, btnFilterTaken)
        }

        btnClearAll.setOnClickListener {
            dbHelper.clearAllHistory(userEmail)
            loadHistory("all") // Refresh the view
        }
    }

    private fun loadHistory(filter: String) {
        val historyList = when (filter) {
            "taken" -> dbHelper.getFilteredHistory("Taken", userEmail)
            "skipped" -> dbHelper.getFilteredHistory("Skipped", userEmail)
            else -> dbHelper.getAllHistory(userEmail)
        }
        // Use the adapter's updateData method to process and display the list
        historyAdapter.updateData(historyList)
    }

    private fun updateFilterButtons(selected: TextView, unselected1: TextView,
```

```
unselected2: TextView) {
        selected.setBackgroundResource(R.drawable.filter_button_selected)
        selected.setTextColor(resources.getColor(android.R.color.white, null))

        unselected1.setBackgroundResource(R.drawable.filter_button_unselected)
        unselected1.setTextColor(resources.getColor(R.color.text_secondary, null))

        unselected2.setBackgroundResource(R.drawable.filter_button_unselected)
        unselected2.setTextColor(resources.getColor(R.color.text_secondary, null))
    }
}
```

HistoryAdapter.kt file

```
package com.example.smartremainder

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import java.text.SimpleDateFormat
import java.util.Date
import java.util.Locale

class HistoryAdapter(private var historyItems: List<HistoryItem>) :
RecyclerView.Adapter<RecyclerView.ViewHolder>() {

    companion object {
        private const val TYPE_DATE = 0
        private const val TYPE_EVENT = 1
    }

    // Sealed class to represent either a date or a history event
    sealed class HistoryItem {
        data class DateItem(val date: String) : HistoryItem()
        data class EventItem(val event: MedicationHistory) : HistoryItem()
    }

    override fun getItemViewType(position: Int): Int {
        return when (historyItems[position]) {
            is HistoryItem.DateItem -> TYPE_DATE
            is HistoryItem.EventItem -> TYPE_EVENT
```

```
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
RecyclerView.ViewHolder {
        return if (viewType == TYPE_DATE) {
            val view =
LayoutInflater.from(parent.context).inflate(R.layout.item_history_date, parent, false)
            DateViewHolder(view)
        } else {
            val view =
LayoutInflater.from(parent.context).inflate(R.layout.item_history_event, parent, false)
            EventViewHolder(view)
        }
    }

    override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {
        when (val item = historyItems[position]) {
            is HistoryItem.DateItem -> (holder as DateViewHolder).bind(item.date)
            is HistoryItem.EventItem -> (holder as EventViewHolder).bind(item.event)
        }
    }

    override fun getItemCount(): Int = historyItems.size

    fun updateData(newHistory: List<MedicationHistory>) {
        val dateFormat = SimpleDateFormat("yyyy-MM-dd", Locale.getDefault())
        val groupedData = newHistory.groupBy { dateFormat.format(Date(it.timestamp))
}
        val items = mutableListOf<HistoryItem>()
        for ((date, events) in groupedData) {
            items.add(HistoryItem.DateItem(date))
            events.forEach { event -> items.add(HistoryItem.EventItem(event)) }
        }
        this.historyItems = items
        notifyDataSetChanged() // Consider using DiffUtil for better performance
    }

    // ViewHolder for Date Headers
    class DateViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        private val tvDate: TextView = itemView.findViewById(R.id.tvHistoryDate)
        fun bind(date: String) {
            tvDate.text = date
        }
    }
```

```
    // ViewHolder for History Events
    class EventViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        private val tvName: TextView =
itemView.findViewById(R.id.tvMedicationName)
        private val tvDose: TextView = itemView.findViewById(R.id.tvMedicationDose)
        private val tvTime: TextView = itemView.findViewById(R.id.tvMedicationTime)
        private val tvStatus: TextView =
itemView.findViewById(R.id.tvMedicationStatus)

        fun bind(history: MedicationHistory) {
            val timeFormat = SimpleDateFormat("HH:mm", Locale.getDefault())
            tvName.text = history.medicationName
            tvDose.text = history.dosage
            tvTime.text = timeFormat.format(Date(history.timestamp))
            tvStatus.text = history.status
            // You might want to set a color based on the status as well
        }
    }
}
```

Login.kt file

```
package com.example.smartremainder

import android.content.Intent
import android.os.Bundle
import android.util.Patterns
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.smartremainder.databinding.ActivityLoginBinding

class Login : AppCompatActivity() {

    private lateinit var binding: ActivityLoginBinding
    private lateinit var dbHelper: SignupSQL

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityLoginBinding.inflate(layoutInflater)
        setContentView(binding.root)

        dbHelper = SignupSQL(this)
```

```
        binding.btnLogin.setOnClickListener {
            if (validateForm()) {
                val email = binding.etEmail.text.toString().trim()
                val password = binding.etPassword.text.toString().trim()

                if (dbHelper.checkLogin(email, password)) {
                    Toast.makeText(this, "Login Successful!",
Toast.LENGTH_SHORT).show()
                    val intent = Intent(this, MainActivity::class.java)
                    intent.putExtra("USER_EMAIL", email)
                    startActivity(intent)
                    finish()
                } else {
                    Toast.makeText(this, "Invalid email or password",
Toast.LENGTH_SHORT).show()
                }
            }
        }

        binding.tvSignUp.setOnClickListener {
            val intent = Intent(this, Signukt::class.java)
            startActivity(intent)
        }
    }

    private fun validateForm(): Boolean {
        var isValid = true

        val email = binding.etEmail.text.toString().trim()
        val password = binding.etPassword.text.toString().trim()

        // Email validation
        if (email.isEmpty()) {
            binding.tilEmail.error = "Email cannot be empty"
            isValid = false
        } else if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            binding.tilEmail.error = "Enter a valid email address"
            isValid = false
        } else {
            binding.tilEmail.error = null
        }

        // Password validation
        if (password.isEmpty()) {
            binding.tilPassword.error = "Password cannot be empty"
```

```
            isValid = false
        } else {
            binding.tilPassword.error = null
        }

        return isValid
    }
}
```

MainActivity.kt file

```
package com.example.smartremainder

import android.Manifest
import android.app.Activity
import android.app.AlarmManager
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.content.IntentFilter
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Build
import android.os.Bundle
import android.provider.Settings
import android.util.Log
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.ContextCompat
import androidx.recyclerview.widget.LinearLayoutManager
import com.bumptech.glide.Glide
import com.example.smartremainder.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding
    private lateinit var dbHelper: DatabaseHelper
    private lateinit var userDbHelper: SignupSQL
    private lateinit var medicationAdapter: MedicationAdapter
    private lateinit var alarmScheduler: AlarmScheduler
    private var permissionsRequested = false
    private var userEmail: String? = null
```

```
    private val requestPermissionLauncher =
        registerForActivityResult(ActivityResultContracts.RequestMultiplePermissions()) {
permissions ->
            permissionsRequested = true
        }

    private val profileLauncher =
registerForActivityResult(ActivityResultContracts.StartActivityForResult()) { result ->
        if (result.resultCode == Activity.RESULT_OK) {
            val imageUriString =
result.data?.getStringExtra("UPDATED_PROFILE_IMAGE_URI")
            if (imageUriString != null) {
                val imageUri = Uri.parse(imageUriString)
                Glide.with(this)
                    .load(imageUri)
                    .circleCrop()
                    .into(binding.ivProfile)
            }
        }
    }

    private val medicationUpdateReceiver = object : BroadcastReceiver() {
        override fun onReceive(context: Context?, intent: Intent?) {
            if (intent?.action == "com.example.smartremainder.MEDICATION_UPDATE") {
                Log.d("MainActivity", "Received medication update broadcast. Refreshing
progress.")
                updateDailyProgress()
                loadMedications()
            }
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        dbHelper = DatabaseHelper(this)
        userDbHelper = SignupSQL(this)
        alarmScheduler = AlarmScheduler(this)

        userEmail = intent.getStringExtra("USER_EMAIL")

        setupQuickActions()
        setupRecyclerView()
        loadUserProfile()
```

```
        val filter = IntentFilter("com.example.smartremainder.MEDICATION_UPDATE")
        ContextCompat.registerReceiver(this, medicationUpdateReceiver, filter,
ContextCompat.RECEIVER_NOT_EXPORTED)
    }

    override fun onResume() {
        super.onResume()
        Log.d("MainActivity", "onResume: Refreshing UI and checking permissions.")
        if (!permissionsRequested) checkAndRequestPermissions()
        loadMedications()
        loadUserProfile()
        updateDailyProgress()
    }

    override fun onDestroy() {
        super.onDestroy()
        unregisterReceiver(medicationUpdateReceiver)
    }

    private fun setupQuickActions() {
        binding.quickActions.llAddMedication.setOnClickListener {
            startActivity(Intent(this, AddMedicineActivity::class.java).apply {
                putExtra("USER_EMAIL", userEmail)
            })
        }
        binding.quickActions.llCalendarView.setOnClickListener {
            startActivity(Intent(this, CalendarActivity::class.java).apply {
                putExtra("USER_EMAIL", userEmail)
            })
        }
        binding.quickActions.llHistoryLog.setOnClickListener {
            startActivity(Intent(this, HistoryActivity::class.java).apply {
                putExtra("USER_EMAIL", userEmail)
            })
        }
        binding.quickActions.llRefillTracker.setOnClickListener {
            startActivity(Intent(this, RefillActivity::class.java).apply {
                putExtra("USER_EMAIL", userEmail)
            })
        }
        binding.quickActions.llUserProfile.setOnClickListener {
            startActivity(Intent(this, data_anaysis::class.java).apply {
                putExtra("USER_EMAIL", userEmail)
            })
        }
```

```
    binding.ivProfile.setOnClickListener {
        val intent = Intent(this, ProfileActivity::class.java)
        intent.putExtra("USER_EMAIL", userEmail)
        profileLauncher.launch(intent)
    }
}

private fun setupRecyclerView() {
    medicationAdapter = MedicationAdapter(emptyList()) { medication ->
        deleteMedication(medication)
    }
    binding.rvMedications.layoutManager = LinearLayoutManager(this)
    binding.rvMedications.adapter = medicationAdapter
}

private fun loadMedications() {
    userEmail?.let {
        val medications = dbHelper.getAllMedications(it)
        medicationAdapter.updateData(medications)
        Log.d("MainActivity", "Loaded ${medications.size} medications for $it")
    }
}

private fun deleteMedication(medication: Medication) {
    userEmail?.let { email ->
        dbHelper.deleteMedication(medication.id, email)
        alarmScheduler.cancel(medication)
        loadMedications()
        updateDailyProgress()
    }
}

private fun loadUserProfile() {
    if (userEmail != null) {
        val user = userDbHelper.getUserByEmail(userEmail!!)
        if (user != null) {
            binding.tvUserName.text = "Welcome, ${user.name}!"
            if (!user.profileImage.isNullOrEmpty()) {
                Glide.with(this)
                    .load(Uri.parse(user.profileImage))
                    .circleCrop()
                    .into(binding.ivProfile)
            }
        } else binding.tvUserName.text = "Welcome, User!"
    } else binding.tvUserName.text = "Welcome, User!"
}
```

```
    private fun checkAndRequestPermissions() {
        val permissionsToRequest = mutableListOf<String>()
        var needsExactAlarmPermission = false

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU &&
            ContextCompat.checkSelfPermission(this,
Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED
        ) {
            permissionsToRequest.add(Manifest.permission.POST_NOTIFICATIONS)
        }

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
            val alarmManager = getSystemService(Context.ALARM_SERVICE) as
AlarmManager
            if (!alarmManager.canScheduleExactAlarms()) needsExactAlarmPermission = true
        }

        when {
            permissionsToRequest.isNotEmpty() ->
requestPermissionLauncher.launch(permissionsToRequest.toTypedArray())
            needsExactAlarmPermission -> {
                permissionsRequested = true

startActivity(Intent(Settings.ACTION_REQUEST_SCHEDULE_EXACT_ALARM).apply {
                    data = Uri.fromParts("package", packageName, null)
                })
            }
        }
    }

    //
    private fun updateDailyProgress() {
        Log.d("MainActivity", "--- Starting Daily Progress Update ---")
        try {
            if (userEmail == null) {
                Log.w("MainActivity", "User not logged in — skipping progress update.")
                binding.circularProgressBar.progress = 0
                binding.tvPercentage.text = getString(R.string.percentage, 0)
                binding.tvDoses.text = "Login required"
                return
            }

            // ✅ Fetch per-user stats
            val totalDoses = dbHelper.getTodayTotalDoses(userEmail!!)
```

```
            val takenDoses = dbHelper.getTodayTakenDoses(userEmail!!)

            Log.i("MainActivity", "User: $userEmail → Total Doses: $totalDoses, Taken:
$takenDoses")

            val percentage = if (totalDoses > 0) (takenDoses * 100) / totalDoses else 0
            val finalPercentage = percentage.coerceIn(0, 100)

            binding.circularProgressBar.progress = finalPercentage
            binding.tvPercentage.text = getString(R.string.percentage, finalPercentage)
            binding.tvDoses.text = getString(R.string.doses_taken, takenDoses, totalDoses)

            Log.i("MainActivity", "UI UPDATED for $userEmail → $finalPercentage%
complete")
        } catch (e: Exception) {
            Log.e("MainActivity", "Error updating daily progress UI: ${e.message}", e)
            binding.circularProgressBar.progress = 0
            binding.tvPercentage.text = getString(R.string.percentage, 0)
            binding.tvDoses.text = getString(R.string.error)
        }
        Log.d("MainActivity", "--- Finished Daily Progress Update ---")
    }
}
```

Medication.kt file

```
package com.example.smartremainder

data class Medication(
    val id: Long = 0,
    val userEmail: String,
    val name: String,
    val dosage: String,
    val frequency: String,
    val duration: Int,
    val startDate: String,
    val reminderEnabled: Boolean,
    val refillEnabled: Boolean,
    val notes: String,
    val currentSupply: Int,
    val maxSupply: Int,
    val image: String?,
```

```
    val times: List<String>
) {
    fun getSupplyStatus(): String {
        val percentage = (currentSupply.toFloat() / maxSupply * 100).toInt()
        return when {
            percentage >= 50 -> "Good"
            percentage >= 20 -> "Low"
            else -> "Critical"
        }
    }

    fun getSupplyPercentage(): Int {
        return if (maxSupply > 0) {
            (currentSupply.toFloat() / maxSupply * 100).toInt()
        } else {
            0
        }
    }
}
```

MedicineActionReceiver.kt file

```
package com.example.smartremainder

import android.app.NotificationManager
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.util.Log

class MedicationActionReceiver : BroadcastReceiver() {

    companion object {
        const val EXTRA_MEDICATION_ID = "MEDICATION_ID"
        const val EXTRA_USER_EMAIL = "USER_EMAIL"
        const val EXTRA_ACTION = "ACTION_TYPE"
        const val ACTION_TAKEN = "TAKEN"
        const val ACTION_SKIPPED = "SKIPPED"
        const val NOTIFICATION_ID = "NOTIFICATION_ID"
        private const val TAG = "MedicationActionReceiver"
    }
```

```
    override fun onReceive(context: Context, intent: Intent) {
        Log.d(TAG, "onReceive started for action: ${intent.action}")

        // Explicitly stop the ringtone service
        val ringtoneIntent = Intent(context, RingtoneService::class.java)
        val stopped = context.stopService(ringtoneIntent)
        Log.d(TAG, "Attempted to stop RingtoneService. Success: $stopped")

        val medId = intent.getLongExtra(EXTRA_MEDICATION_ID, -1)
        val userEmail = intent.getStringExtra(EXTRA_USER_EMAIL)
        val action = intent.getStringExtra(EXTRA_ACTION)
        val notificationId = intent.getIntExtra(NOTIFICATION_ID, -1)

        Log.d(TAG, "Received medId: $medId, userEmail: $userEmail, action: $action,
notificationId: $notificationId")

        if (medId != -1L && userEmail != null && action != null) {
            val dbHelper = DatabaseHelper(context)
            val success = when (action) {
                ACTION_TAKEN -> {
                    Log.d(TAG, "Marking dose as TAKEN")
                    dbHelper.markMostRecentDoseAsTaken(medId, userEmail)
                }
                ACTION_SKIPPED -> {
                    Log.d(TAG, "Marking dose as SKIPPED")
                    dbHelper.markMostRecentDoseAsSkipped(medId, userEmail)
                }
                else -> {
                    Log.w(TAG, "Unknown action: $action")
                    false
                }
            }

            Log.d(TAG, "Database update success: $success")

            if (success) {
                // Dismiss the notification
                val notificationManager =
context.getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
                val idToCancel = if (notificationId != -1) notificationId else medId.toInt()
                notificationManager.cancel(idToCancel)
                Log.d(TAG, "Cancelled notification with ID: $idToCancel")

                // Send broadcast to update UI
                sendUpdateBroadcast(context)
```

```
            Log.d(TAG, "Sent MEDICATION_UPDATE broadcast")
        } else {
            Log.e(TAG, "Failed to update dose status in the database.")
        }
    } else {
        Log.e(TAG, "Received invalid data in intent. Cannot process action.")
    }
  }

  private fun sendUpdateBroadcast(context: Context) {
    val intent = Intent("com.example.smartremainder.MEDICATION_UPDATE")
    intent.setPackage(context.packageName) // Restrict broadcast to this app
    context.sendBroadcast(intent)
  }
}
```

MedicineAdapter.kt file

```
package com.example.smartremainder

import android.graphics.Color
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class MedicationAdapter(
  private var medications: List<Medication> = emptyList(),
  private val onDeleteClick: ((Medication) -> Unit)? = null
) : RecyclerView.Adapter<MedicationAdapter.MedicationViewHolder>() {

  override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
MedicationViewHolder {
    val view = LayoutInflater.from(parent.context).inflate(R.layout.item_medication, parent,
false)
    return MedicationViewHolder(view)
  }

  override fun onBindViewHolder(holder: MedicationViewHolder, position: Int) {
    val medication = medications[position]
    holder.medicationName.text = medication.name
    holder.medicationDosage.text = medication.dosage
    holder.medicationTime.text = medication.times.joinToString(", ")
```

```
        holder.medicationStatus.text = medication.getSupplyStatus()
        // Set color bar based on some logic if available, otherwise default
        // holder.colorBar.setBackgroundColor(Color.parseColor(medication.color))

        holder.itemView.setOnLongClickListener {
            onDeleteClick?.invoke(medication)
            true
        }
    }

    override fun getItemCount() = medications.size

    fun updateData(newMedications: List<Medication>) {
        medications = newMedications
        notifyDataSetChanged()
    }

    fun updateMedications(newMedications: List<Medication>) {
        updateData(newMedications)
    }

    class MedicationViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val medicationName: TextView = itemView.findViewById(R.id.medicationName)
        val medicationDosage: TextView = itemView.findViewById(R.id.medicationDosage)
        val medicationTime: TextView = itemView.findViewById(R.id.medicationTime)
        val medicationStatus: TextView = itemView.findViewById(R.id.medicationStatus)
        val colorBar: View = itemView.findViewById(R.id.colorBar)
    }
}
```

NotificationHelper.kt file

```
package com.example.smartremainder

import android.app.NotificationChannel
import android.app.NotificationManager
import android.app.PendingIntent
import android.content.Context
import android.content.Intent
import android.media.AudioAttributes
import android.media.RingtoneManager
import android.net.Uri
import android.os.Build
import androidx.core.app.NotificationCompat
```

```
class NotificationHelper(private val context: Context) {

    companion object {
        private const val CHANNEL_ID = "medication_reminder_channel"
        private const val CHANNEL_NAME = "Medication Reminders"
    }

    private val notificationManager =
context.getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager

    init {
        createNotificationChannel()
    }

    private fun createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            val channel = NotificationChannel(
                CHANNEL_ID,
                CHANNEL_NAME,
                NotificationManager.IMPORTANCE_HIGH
            ).apply {
                description = "Channel for medication reminder alarms"
                setBypassDnd(true)

                val audioAttributes = AudioAttributes.Builder()
                    .setUsage(AudioAttributes.USAGE_ALARM)
                    .setContentType(AudioAttributes.CONTENT_TYPE_SONIFICATION)
                    .build()
                setSound(null, audioAttributes)
            }
            notificationManager.createNotificationChannel(channel)
        }
    }

    fun showNotification(medication: Medication) {
        val userEmail = medication.userEmail
        val notificationId = medication.id.toInt()
        val currentTime = System.currentTimeMillis()

        val mainIntent = Intent(context, MainActivity::class.java).apply {
            flags = Intent.FLAG_ACTIVITY_NEW_TASK or
Intent.FLAG_ACTIVITY_CLEAR_TASK
        }
        val mainPendingIntent = PendingIntent.getActivity(
            context,
```

```
            notificationId, // A consistent request code for the main tap action
            mainIntent,
            PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTABLE
        )

        // Using medication.id ensures request codes are unique per action per notification
        val takenRequestCode = medication.id.toInt() * 10 + 1
        val skippedRequestCode = medication.id.toInt() * 10 + 2

        // "Taken" button Intent
        val takenIntent = Intent(context, MedicationActionReceiver::class.java).apply {
            // Set a unique action and data URI to ensure the PendingIntent is unique
            action =
"${MedicationActionReceiver.ACTION_TAKEN}_${medication.id}_$currentTime"
            data = Uri.parse("smartremainder://medication/action/${action}")
            putExtra(MedicationActionReceiver.EXTRA_ACTION,
MedicationActionReceiver.ACTION_TAKEN)
            putExtra(MedicationActionReceiver.EXTRA_MEDICATION_ID, medication.id)
            putExtra(MedicationActionReceiver.EXTRA_USER_EMAIL, userEmail)
            putExtra(MedicationActionReceiver.NOTIFICATION_ID, notificationId)
        }
        val takenPendingIntent = PendingIntent.getBroadcast(
            context,
            takenRequestCode,
            takenIntent,
            PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTABLE
        )

        // "Skipped" button Intent
        val skippedIntent = Intent(context, MedicationActionReceiver::class.java).apply {
            // Set a unique action and data URI to ensure the PendingIntent is unique
            action =
"${MedicationActionReceiver.ACTION_SKIPPED}_${medication.id}_$currentTime"
            data = Uri.parse("smartremainder://medication/action/${action}")
            putExtra(MedicationActionReceiver.EXTRA_ACTION,
MedicationActionReceiver.ACTION_SKIPPED)
            putExtra(MedicationActionReceiver.EXTRA_MEDICATION_ID, medication.id)
            putExtra(MedicationActionReceiver.EXTRA_USER_EMAIL, userEmail)
            putExtra(MedicationActionReceiver.NOTIFICATION_ID, notificationId)
        }
        val skippedPendingIntent = PendingIntent.getBroadcast(
            context,
            skippedRequestCode,
            skippedIntent,
            PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTABLE
        )
```

```
        val builder = NotificationCompat.Builder(context, CHANNEL_ID)
            .setSmallIcon(R.drawable.ic_launcher_foreground)
            .setContentTitle("Time for your medication!")
            .setContentText("Take ${medication.name} (${medication.dosage})")
            .setPriority(NotificationCompat.PRIORITY_MAX)
            .setCategory(NotificationCompat.CATEGORY_ALARM)
            .setContentIntent(mainPendingIntent)
            .setAutoCancel(true)
            .addAction(R.drawable.ic_check_circle, "Taken", takenPendingIntent)
            .addAction(R.drawable.ic_cancel, "Skipped", skippedPendingIntent)

        notificationManager.notify(notificationId, builder.build())
    }

    fun cancelNotification(notificationId: Int) {
        notificationManager.cancel(notificationId)
    }
}
```

ProfileActivity.kt file

```
package com.example.smartremainder

import android.app.Activity
import android.content.Context
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.widget.EditText
import android.widget.Toast
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import com.bumptech.glide.Glide
import com.example.smartremainder.databinding.ActivityProfileBinding

class ProfileActivity : AppCompatActivity() {

    private lateinit var binding: ActivityProfileBinding
    private lateinit var dbHelper: UserDatabaseHelper
    private lateinit var mainDbHelper: SignupSQL // Permanent database
    private var currentUser: UserProfile? = null
    private var userEmail: String? = null
```

```
    private val pickImageLauncher =
registerForActivityResult(ActivityResultContracts.StartActivityForResult()) { result ->
        if (result.resultCode == Activity.RESULT_OK) {
            val imageUri = result.data?.data
            if (imageUri != null) {
                // Request persistent permissions for the selected image
                val takeFlags: Int = Intent.FLAG_GRANT_READ_URI_PERMISSION
                contentResolver.takePersistableUriPermission(imageUri, takeFlags)

                currentUser?.let {
                    // Save to the session database
                    val updatedUser = it.copy(profileImage = imageUri.toString())
                    dbHelper.insertOrUpdateUser(updatedUser)

                    // Save to the permanent database
                    mainDbHelper.updateUserProfileImage(it.email, imageUri.toString())

                    // Set the result to be sent back to MainActivity
                    val resultIntent = Intent()
                    resultIntent.putExtra("UPDATED_PROFILE_IMAGE_URI",
imageUri.toString())
                    setResult(Activity.RESULT_OK, resultIntent)

                    loadUserProfile()
                }
            }
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityProfileBinding.inflate(layoutInflater)
        setContentView(binding.root)

        dbHelper = UserDatabaseHelper(this)
        mainDbHelper = SignupSQL(this)

        // Retrieve the user's email from the intent
        userEmail = intent.getStringExtra("USER_EMAIL")

        if (userEmail == null) {
            Toast.makeText(this, "Error: User not identified.", Toast.LENGTH_LONG).show()
            finish()
            return
        }
```

```
    setupToolbar()
    loadUserProfile()
    setupButtons()

    binding.ivEditProfileImage.setOnClickListener {
        openGallery()
    }
}

private fun openGallery() {
    // Use ACTION_OPEN_DOCUMENT to get a persistable URI
    val intent = Intent(Intent.ACTION_OPEN_DOCUMENT).apply {
        addCategory(Intent.CATEGORY_OPENABLE)
        type = "image/*"
    }
    pickImageLauncher.launch(intent)
}

private fun setupToolbar() {
    binding.toolbar.setNavigationOnClickListener {
        finish()
    }
}

private fun loadUserProfile() {
    // Load the user from the permanent database using the email
    val userFromDb = mainDbHelper.getUserByEmail(userEmail!!)

    if (userFromDb != null) {
        // Create a UserProfile object to hold the data for the current session
        currentUser = userFromDb
        // Cache the user profile in the session helper
        dbHelper.insertOrUpdateUser(currentUser!!)
        displayUserProfile(currentUser!!)
    } else {
        Toast.makeText(this, "User profile not found.", Toast.LENGTH_LONG).show()
        finish()
    }
}

private fun displayUserProfile(user: UserProfile) {
    with(binding) {
        tvUserName.text = user.name
        tvUsername.text = "@${user.name}" // Using name as a substitute for username
        tvEmail.text = user.email
        tvPhone.text = "Phone: ${user.phone}"
```

```
        if (!user.profileImage.isNullOrEmpty()) {
            Glide.with(this@ProfileActivity)
                .load(Uri.parse(user.profileImage))
                .circleCrop()
                .into(ivProfilePicture)
        } else {
            // Set a default image if no profile picture is selected
            Glide.with(this@ProfileActivity)
                .load(R.drawable.ic_user)
                .circleCrop()
                .into(ivProfilePicture)
        }
    }
}

private fun setupButtons() {
    binding.btnEditProfile.setOnClickListener {
        openEditProfileDialog()
    }

    binding.btnLogout.setOnClickListener {
        showLogoutConfirmation()
    }
}

private fun openEditProfileDialog() {
    val dialogView = layoutInflater.inflate(R.layout.dialog_edit_profile, null)
    val etName = dialogView.findViewById<EditText>(R.id.etName)
    val etUsername = dialogView.findViewById<EditText>(R.id.etUsername)
    val etEmail = dialogView.findViewById<EditText>(R.id.etEmail)
    val etPhone = dialogView.findViewById<EditText>(R.id.etPhone)

    currentUser?.let { user ->
        etName.setText(user.name)
        etUsername.setText(user.name) // Using name as a substitute for username
        etEmail.setText(user.email)
        etPhone.setText(user.phone)
    }

    AlertDialog.Builder(this)
        .setTitle("Edit Profile")
        .setView(dialogView)
        .setPositiveButton("Save") { _, _ ->
            val updatedName = etName.text.toString()
            val updatedPhone = etPhone.text.toString()
```

```
            currentUser?.let {
                mainDbHelper.updateUserProfile(it.email, updatedName, updatedPhone)
                val updatedUser = it.copy(name = updatedName, phone = updatedPhone)
                dbHelper.insertOrUpdateUser(updatedUser)

                loadUserProfile() // Reload to reflect changes
                Toast.makeText(this, "Profile updated successfully",
Toast.LENGTH_SHORT).show()
            }
        }
        .setNegativeButton("Cancel", null)
        .create()
        .show()
    }

    private fun showLogoutConfirmation() {
        AlertDialog.Builder(this)
            .setTitle("Logout")
            .setMessage("Are you sure you want to logout?")
            .setPositiveButton("Logout") { _, _ ->
                performLogout()
            }
            .setNegativeButton("Cancel", null)
            .create()
            .show()
    }

    private fun performLogout() {
        dbHelper.deleteUser()

        val prefs = getSharedPreferences("user_prefs", Context.MODE_PRIVATE)
        prefs.edit().clear().apply()

        val intent = Intent(this, WelComePage::class.java)
        intent.flags = Intent.FLAG_ACTIVITY_NEW_TASK or
Intent.FLAG_ACTIVITY_CLEAR_TASK
        startActivity(intent)

        Toast.makeText(this, "Logged out successfully", Toast.LENGTH_SHORT).show()
        finish()
    }
}
```

RefilActivity.kt file

```
package com.example.smartremainder

import android.os.Bundle
import android.widget.ImageButton
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.smartremainder.databinding.ActivityRefillBinding

class RefillActivity : AppCompatActivity() {

    private lateinit var binding: ActivityRefillBinding
    private lateinit var dbHelper: DatabaseHelper
    private lateinit var adapter: RefillAdapter
    private lateinit var userEmail: String

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityRefillBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val backButton = findViewById<ImageButton>(R.id.backButton)
        backButton.setOnClickListener {
            finish()
        }

        dbHelper = DatabaseHelper(this)
        userEmail = intent.getStringExtra("USER_EMAIL") ?: ""

        setupRecyclerView()
        loadMedicines()
    }

    private fun setupRecyclerView() {
        adapter = RefillAdapter(emptyList()) { medicine ->
            showRefillDialog(medicine)
        }

        binding.refillRecyclerView.apply {
            layoutManager = LinearLayoutManager(this@RefillActivity)
            adapter = this@RefillActivity.adapter
        }
    }
```

```
    private fun loadMedicines() {
        val medicines = dbHelper.getAllMedications(userEmail)
        adapter.updateData(medicines)
    }

    private fun showRefillDialog(medicine: Medication) {
        val dialog = AlertDialog.Builder(this)
            .setTitle("Record Refill")
            .setMessage("Refill ${medicine.name} to maximum supply (${medicine.maxSupply}
units)?")
            .setPositiveButton("Refill") { _, _ ->
                val refillAmount = medicine.maxSupply - medicine.currentSupply
                dbHelper.recordRefill(medicine.id, userEmail, refillAmount)
                loadMedicines()
                Toast.makeText(this, "Refill recorded successfully",
Toast.LENGTH_SHORT).show()
            }
            .setNegativeButton("Cancel", null)
            .create()
        dialog.show()
    }
}
```

RefillAdapter.kt file

```
package com.example.smartremainder

import android.graphics.Color
import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.example.smartremainder.databinding.ItemRefillMedicineBinding

class RefillAdapter(
    private var medicines: List<Medication>,
    private val onRefillClick: (Medication) -> Unit
) : RecyclerView.Adapter<RefillAdapter.RefillViewHolder>() {

    inner class RefillViewHolder(val binding: ItemRefillMedicineBinding) :
        RecyclerView.ViewHolder(binding.root)

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RefillViewHolder {
        val binding = ItemRefillMedicineBinding.inflate(
```

```
            LayoutInflater.from(parent.context), parent, false
        )
        return RefillViewHolder(binding)
    }

    override fun onBindViewHolder(holder: RefillViewHolder, position: Int) {
        val medicine = medicines[position]
        with(holder.binding) {
            tvMedicineName.text = medicine.name
            tvDosage.text = medicine.dosage
            tvCurrentSupply.text = "${medicine.currentSupply} units"
            tvSupplyStatus.text = medicine.getSupplyStatus()

            // Set progress bar
            progressBar.progress = medicine.getSupplyPercentage()
            progressBar.max = 100

            // Set status color and color indicator
            val statusColor = when (medicine.getSupplyStatus()) {
                "Good" -> Color.parseColor("#4CAF50")
                "Low" -> Color.parseColor("#FF9800")
                else -> Color.parseColor("#F44336")
            }
            tvSupplyStatus.setTextColor(statusColor)
            colorIndicator.setBackgroundColor(statusColor)

            btnRecordRefill.setOnClickListener {
                onRefillClick(medicine)
            }
        }
    }

    override fun getItemCount() = medicines.size

    fun updateData(newMedicines: List<Medication>) {
        medicines = newMedicines
        notifyDataSetChanged()
    }
}
```

RingToneService.kt file

```
package com.example.smartremainder

import android.app.Service
import android.content.Intent
```

```
import android.media.Ringtone
import android.media.RingtoneManager
import android.os.IBinder

class RingtoneService : Service() {

    private var ringtone: Ringtone? = null

    override fun onBind(intent: Intent?): IBinder? {
        return null
    }

    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        if (ringtone == null) {
            val ringtoneUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM)
            ringtone = RingtoneManager.getRingtone(this, ringtoneUri)
        }

        if (ringtone?.isPlaying == false) {
            ringtone?.play()
        }

        return START_NOT_STICKY
    }

    override fun onDestroy() {
        super.onDestroy()
        ringtone?.stop()
        ringtone = null
    }
}
```

signup.kt file

```
package com.example.smartremainder

import android.content.Intent
import android.os.Bundle
import android.util.Patterns
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.smartremainder.databinding.ActivitySignuktBinding
import java.util.regex.Pattern
```

```
class Signukt : AppCompatActivity() {

    private lateinit var binding: ActivitySignuktBinding
    private lateinit var dbHelper: SignupSQL

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivitySignuktBinding.inflate(layoutInflater)
        setContentView(binding.root)

        dbHelper = SignupSQL(this)

        binding.btnSignUp.setOnClickListener {
            if (validateForm()) {
                val name = binding.etFullName.text.toString().trim()
                val email = binding.etEmail.text.toString().trim()
                val phone = binding.etPhone.text.toString().trim()
                val password = binding.etPassword.text.toString().trim()

                if (dbHelper.checkUserExists(email)) {
                    Toast.makeText(this, "User with this email already exists",
Toast.LENGTH_SHORT).show()
                } else {
                    val isInserted = dbHelper.insertUser(name, email, phone, password)
                    if (isInserted) {
                        Toast.makeText(this, "Account created successfully! Please login.",
Toast.LENGTH_SHORT).show()
                        val intent = Intent(this, Login::class.java)
                        startActivity(intent)
                        finish()
                    } else {
                        Toast.makeText(this, "Failed to create account",
Toast.LENGTH_SHORT).show()
                    }
                }
            }
        }

        binding.tvLogin.setOnClickListener {
            startActivity(Intent(this, Login::class.java))
        }
    }

    private fun validateForm(): Boolean {
        var isValid = true
```

```
        val name = binding.etFullName.text.toString().trim()
        val email = binding.etEmail.text.toString().trim()
        val phone = binding.etPhone.text.toString().trim()
        val password = binding.etPassword.text.toString().trim()
        val confirmPassword = binding.etConfirmPassword.text.toString().trim()

        // Name validation
        if (name.length < 3) {
            binding.tilFullName.error = "Name must be at least 3 characters"
            isValid = false
        } else {
            binding.tilFullName.error = null
        }

        // Email validation
        if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            binding.tilEmail.error = "Enter a valid email address"
            isValid = false
        } else {
            binding.tilEmail.error = null
        }

        // Phone validation
        if (phone.length != 10 || !phone.all { it.isDigit() }) {
            binding.tilPhone.error = "Enter a valid 10-digit phone number"
            isValid = false
        } else {
            binding.tilPhone.error = null
        }

        // Password validation
        val passwordPattern = Pattern.compile(
            "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[@#\$%^&+=!])(?=\\S+$).{8,}$"
        )
        if (!passwordPattern.matcher(password).matches()) {
            binding.tilPassword.error = "Password must be 8+ chars with upper, lower, digit, and
special char"
            isValid = false
        } else {
            binding.tilPassword.error = null
        }

        // Confirm password validation
        if (password != confirmPassword) {
            binding.tilConfirmPassword.error = "Passwords do not match"
            isValid = false
```

```
        } else {
            binding.tilConfirmPassword.error = null
        }

        // Terms and conditions validation
        if (!binding.cbTerms.isChecked) {
            binding.cbTerms.error = "You must accept the Terms and Conditions"
            isValid = false
        } else {
            binding.cbTerms.error = null
        }

        return isValid
    }
}
```

SignUpSql.kt file

```
package com.example.smartremainder

import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class SignupSQL(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, null,
DATABASE_VERSION) {

    companion object {
        private const val DATABASE_NAME = "UserDatabase.db"
        private const val DATABASE_VERSION = 2 // Incremented the version
        private const val TABLE_NAME = "users"
        private const val COLUMN_ID = "id"
        private const val COLUMN_NAME = "name"
        private const val COLUMN_EMAIL = "email"
        private const val COLUMN_PHONE = "phone"
        private const val COLUMN_PASSWORD = "password"
        private const val COLUMN_PROFILE_IMAGE = "profile_image" // New Column
    }

    override fun onCreate(db: SQLiteDatabase?) {
        val createTable = ("CREATE TABLE $TABLE_NAME (" +
                "$COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
```

```
            "$COLUMN_NAME TEXT, " +
            "$COLUMN_EMAIL TEXT UNIQUE, " +
            "$COLUMN_PHONE TEXT, " +
            "$COLUMN_PASSWORD TEXT, " +
            "$COLUMN_PROFILE_IMAGE TEXT)") // Added new column
        db?.execSQL(createTable)
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
        if (oldVersion < 2) {
            db?.execSQL("ALTER TABLE $TABLE_NAME ADD COLUMN
$COLUMN_PROFILE_IMAGE TEXT")
        }
    }

    fun insertUser(name: String, email: String, phone: String, password: String): Boolean {
        val db = writableDatabase
        val values = ContentValues().apply {
            put(COLUMN_NAME, name)
            put(COLUMN_EMAIL, email)
            put(COLUMN_PHONE, phone)
            put(COLUMN_PASSWORD, password)
        }

        val result = db.insert(TABLE_NAME, null, values)
        return result != -1L
    }

    fun checkUserExists(email: String): Boolean {
        val db = readableDatabase
        val query = "SELECT * FROM $TABLE_NAME WHERE $COLUMN_EMAIL = ?"
        val cursor = db.rawQuery(query, arrayOf(email))
        val exists = cursor.count > 0
        cursor.close()
        return exists
    }

    fun checkLogin(email: String, password: String): Boolean {
        val db = readableDatabase
        val query = "SELECT * FROM $TABLE_NAME WHERE $COLUMN_EMAIL = ?
AND $COLUMN_PASSWORD = ?"
        val cursor = db.rawQuery(query, arrayOf(email, password))
        val isValid = cursor.count > 0
        cursor.close()
        return isValid
    }
```

```
    fun getUserByEmail(email: String): UserProfile? {
        val db = readableDatabase
        val query = "SELECT * FROM $TABLE_NAME WHERE $COLUMN_EMAIL = ?"
        val cursor = db.rawQuery(query, arrayOf(email))
        var user: UserProfile? = null
        if (cursor.moveToFirst()) {
            val id = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_ID))
            val name = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_NAME))
            val phone = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_PHONE))
            val profileImageIndex = cursor.getColumnIndex(COLUMN_PROFILE_IMAGE)
            val profileImage = if (profileImageIndex != -1 && !cursor.isNull(profileImageIndex))
cursor.getString(profileImageIndex) else null
            // For username, which is not in this DB, we'll use name as a default
            user = UserProfile(id, name, name, email, phone, profileImage)
        }
        cursor.close()
        return user
    }

    fun updateUserProfileImage(email: String, imageUri: String): Int {
        val db = writableDatabase
        val values = ContentValues().apply {
            put(COLUMN_PROFILE_IMAGE, imageUri)
        }
        return db.update(TABLE_NAME, values, "$COLUMN_EMAIL = ?", arrayOf(email))
    }

    fun updateUserProfile(email: String, name: String, phone: String): Int {
        val db = writableDatabase
        val values = ContentValues().apply {
            put(COLUMN_NAME, name)
            put(COLUMN_PHONE, phone)
        }
        return db.update(TABLE_NAME, values, "$COLUMN_EMAIL = ?", arrayOf(email))
    }
}
```

SplashScreen.kt file

```
package com.example.smartremainder

import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import androidx.appcompat.app.AppCompatActivity

class Splashscreen : AppCompatActivity() {

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

        // Delay for 2.5 seconds, then open Login screen
        Handler(Looper.getMainLooper()).postDelayed({
            val intent = Intent(this, WelComePage::class.java)
            startActivity(intent)
            finish()
        }, 2500) // 2500 ms = 2.5 seconds
    }
}
```

UserDataBaseHelper.kt file

```
package com.example.smartremainder

import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class UserDatabaseHelper(context: Context) : SQLiteOpenHelper(context,
DATABASE_NAME, null, DATABASE_VERSION) {

    companion object {
        private const val DATABASE_NAME = "UserProfile.db"
        private const val DATABASE_VERSION = 1
        private const val TABLE_USER = "user_profile"

        private const val COL_ID = "id"
        private const val COL_NAME = "name"
        private const val COL_USERNAME = "username"
        private const val COL_EMAIL = "email"
```

```
    private const val COL_PHONE = "phone"
    private const val COL_PROFILE_IMAGE = "profile_image"
  }

  override fun onCreate(db: SQLiteDatabase) {
    val createTable = """
      CREATE TABLE $TABLE_USER (
        $COL_ID INTEGER PRIMARY KEY AUTOINCREMENT,
        $COL_NAME TEXT NOT NULL,
        $COL_USERNAME TEXT NOT NULL,
        $COL_EMAIL TEXT NOT NULL,
        $COL_PHONE TEXT NOT NULL,
        $COL_PROFILE_IMAGE TEXT
      )
    """.trimIndent()
    db.execSQL(createTable)
  }

  override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {
    db.execSQL("DROP TABLE IF EXISTS $TABLE_USER")
    onCreate(db)
  }

  fun getUserProfile(): UserProfile? {
    val db = readableDatabase
    val cursor = db.rawQuery("SELECT * FROM $TABLE_USER LIMIT 1", null)

    var userProfile: UserProfile? = null
    if (cursor.moveToFirst()) {
      userProfile = UserProfile(
        id = cursor.getInt(cursor.getColumnIndexOrThrow(COL_ID)),
        name = cursor.getString(cursor.getColumnIndexOrThrow(COL_NAME)),
        username =
cursor.getString(cursor.getColumnIndexOrThrow(COL_USERNAME)),
        email = cursor.getString(cursor.getColumnIndexOrThrow(COL_EMAIL)),
        phone = cursor.getString(cursor.getColumnIndexOrThrow(COL_PHONE)),
        profileImage =
cursor.getString(cursor.getColumnIndexOrThrow(COL_PROFILE_IMAGE))
      )
    }
    cursor.close()
    return userProfile
  }

  fun insertOrUpdateUser(userProfile: UserProfile): Long {
    val db = writableDatabase
```

```
        val values = ContentValues().apply {
            put(COL_NAME, userProfile.name)
            put(COL_USERNAME, userProfile.username)
            put(COL_EMAIL, userProfile.email)
            put(COL_PHONE, userProfile.phone)
            put(COL_PROFILE_IMAGE, userProfile.profileImage)
        }

        // Check if user exists
        val cursor = db.rawQuery("SELECT $COL_ID FROM $TABLE_USER LIMIT 1", null)
        val result = if (cursor.moveToFirst()) {
            val id = cursor.getInt(0)
            db.update(TABLE_USER, values, "$COL_ID = ?", arrayOf(id.toString())).toLong()
        } else {
            db.insert(TABLE_USER, null, values)
        }
        cursor.close()
        return result
    }

    fun deleteUser(): Int {
        val db = writableDatabase
        return db.delete(TABLE_USER, null, null)
    }
}
```

welcomePage.kt file

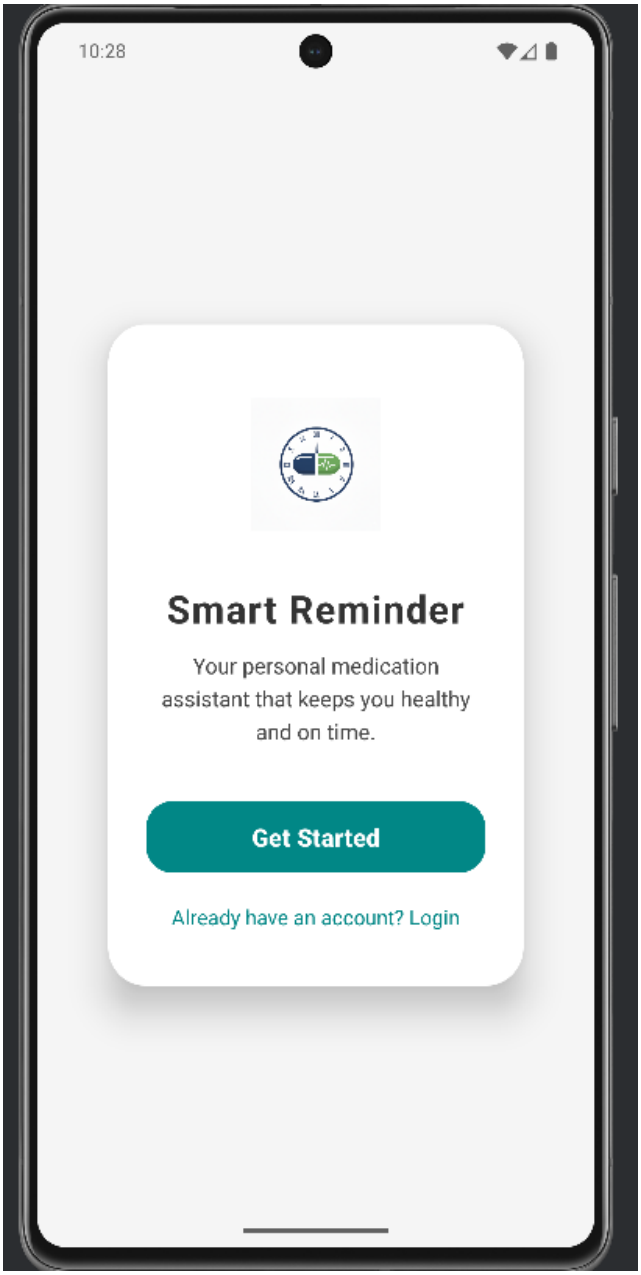
```
package com.example.smartremainder

import android.content.Intent
import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import com.example.smartremainder.databinding.ActivityWelComePageBinding

class WelComePage : AppCompatActivity() {

    private lateinit var binding: ActivityWelComePageBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        binding = ActivityWelComePageBinding.inflate(layoutInflater)
```

```
        setContentView(binding.root)

        val dbHelper = SignupSQL(this)
        val db = dbHelper.readableDatabase  // Opens DB connection


        // Navigate to SignUpActivity
        binding.btnGetStarted.setOnClickListener {
            val intent = Intent(this, Signukt::class.java)
            startActivity(intent)
        }

        // Navigate to LoginActivity
        binding.loginText.setOnClickListener {
            val intent = Intent(this, Login::class.java)
            startActivity(intent)
        }
    }
}
```

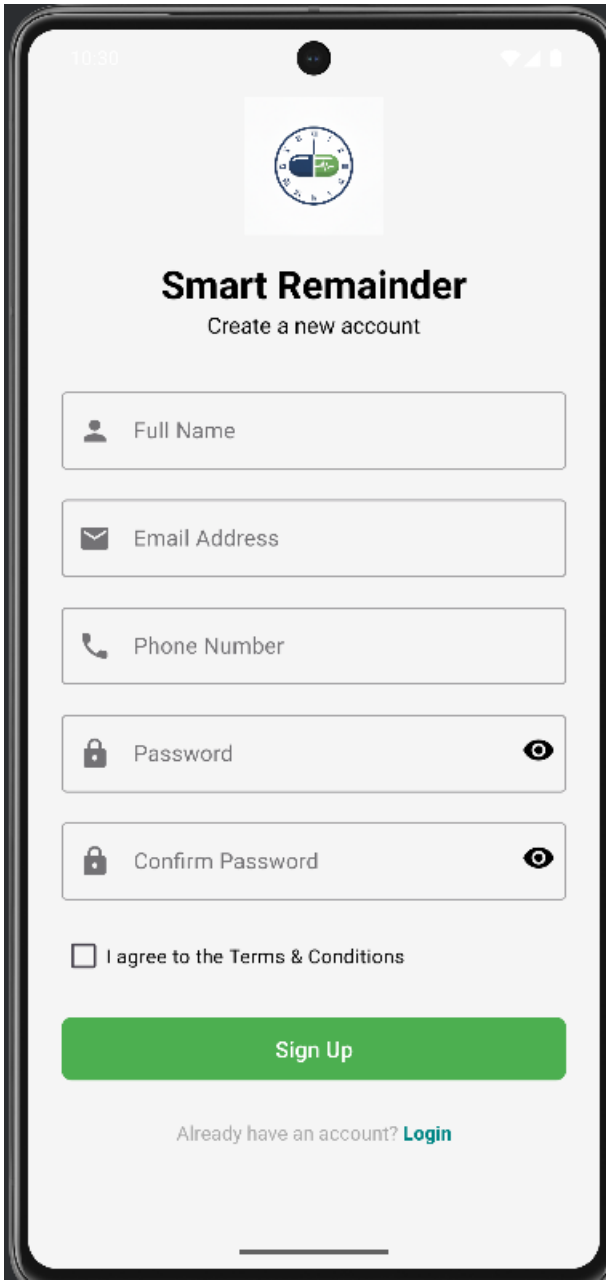
Result:

Splash Screen

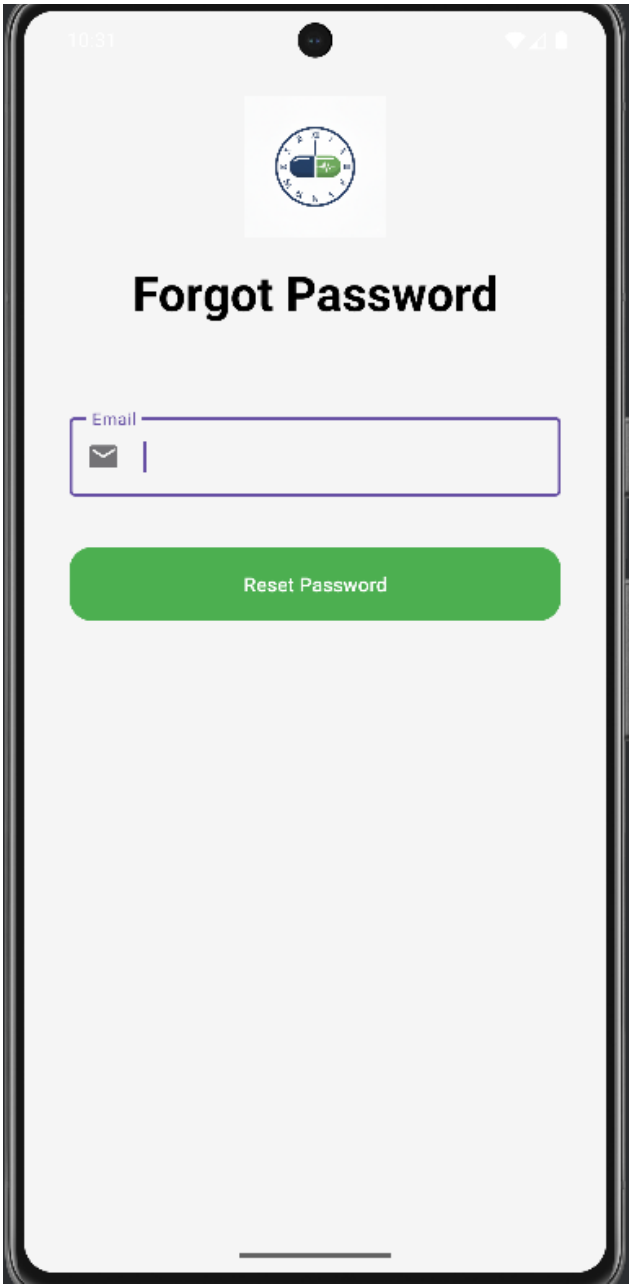
Login Page



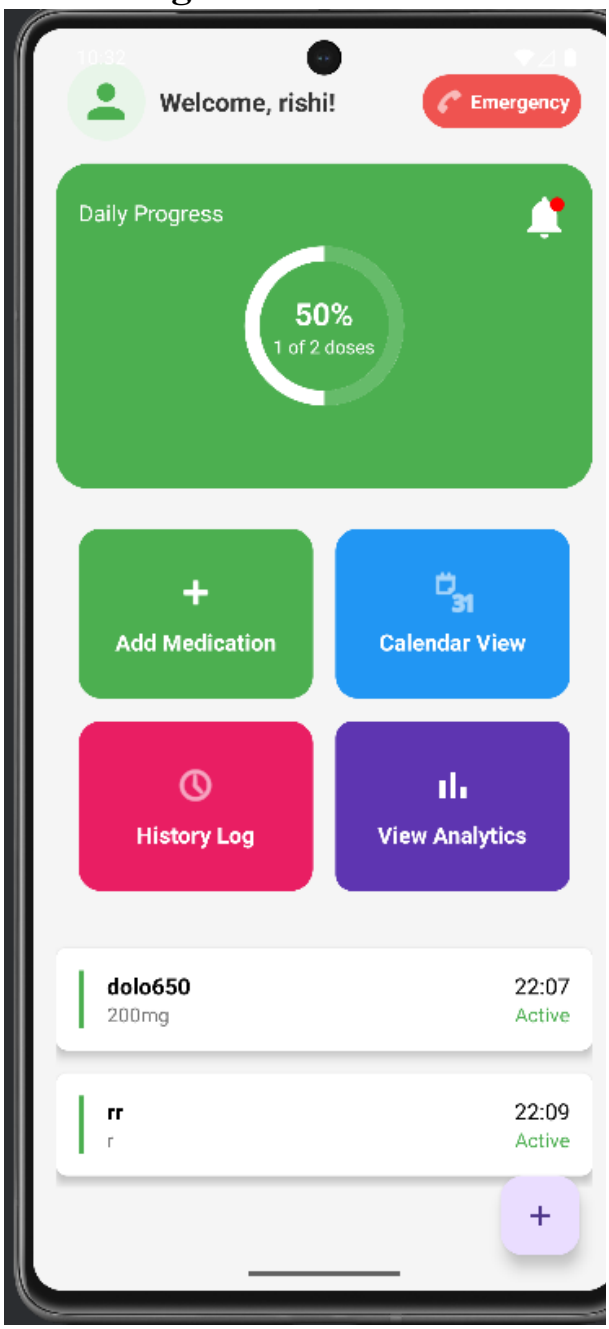
Sign Up page



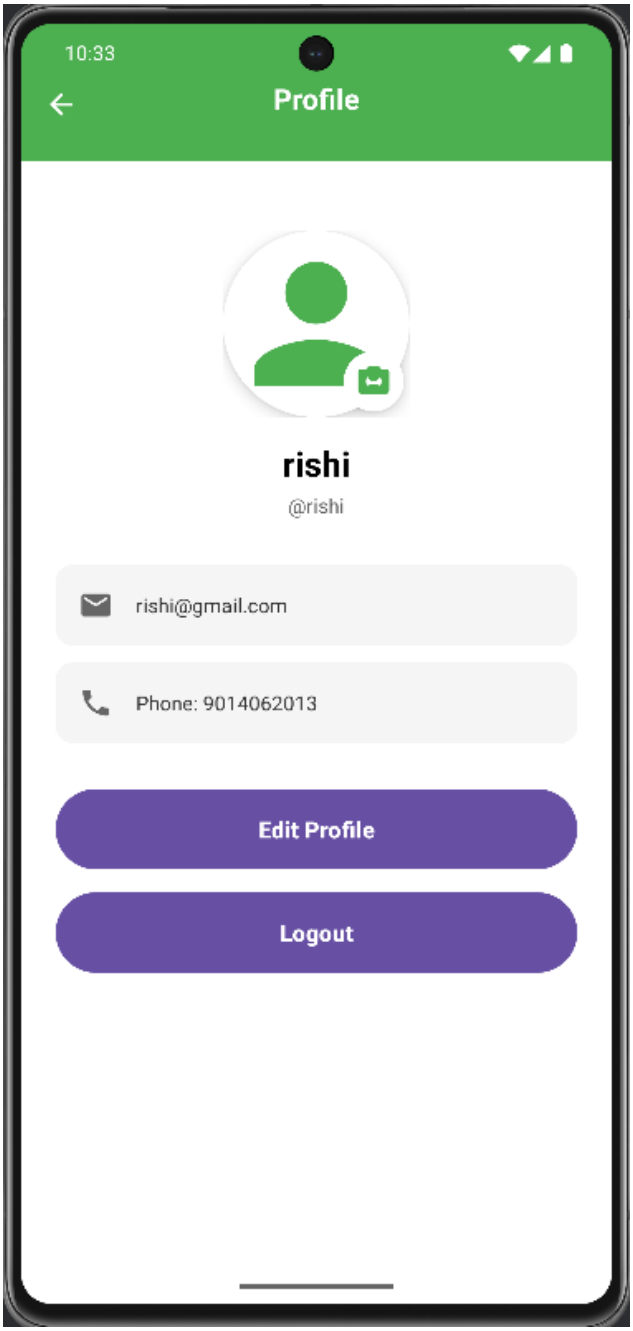
Forgot Page



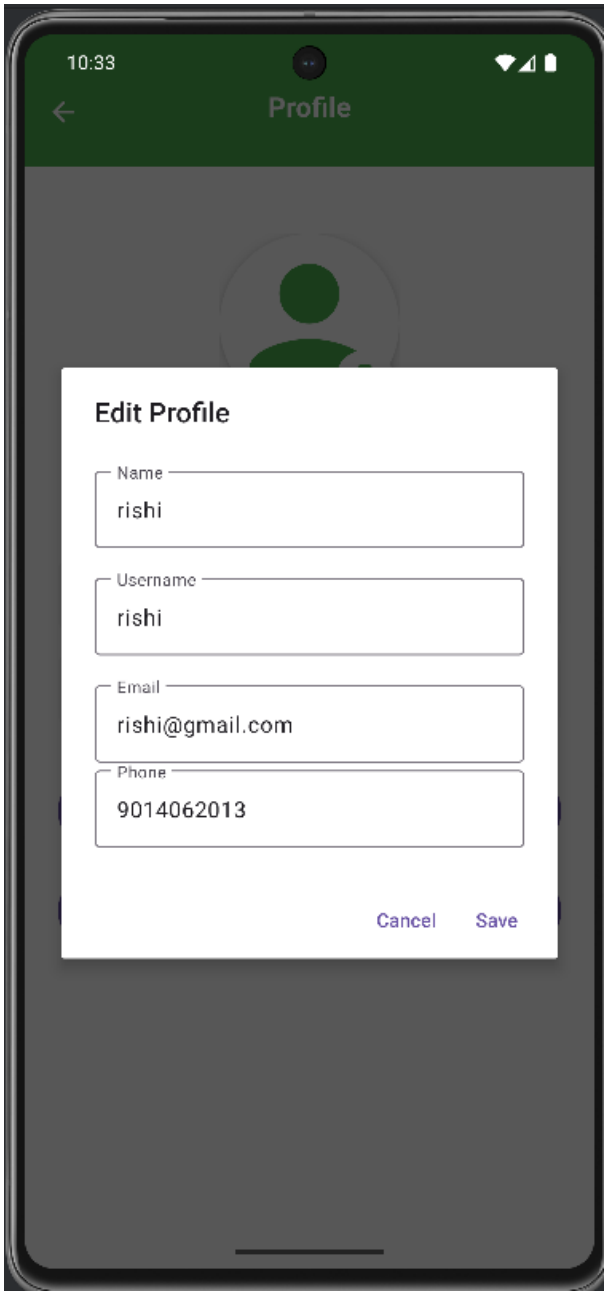
Home Page



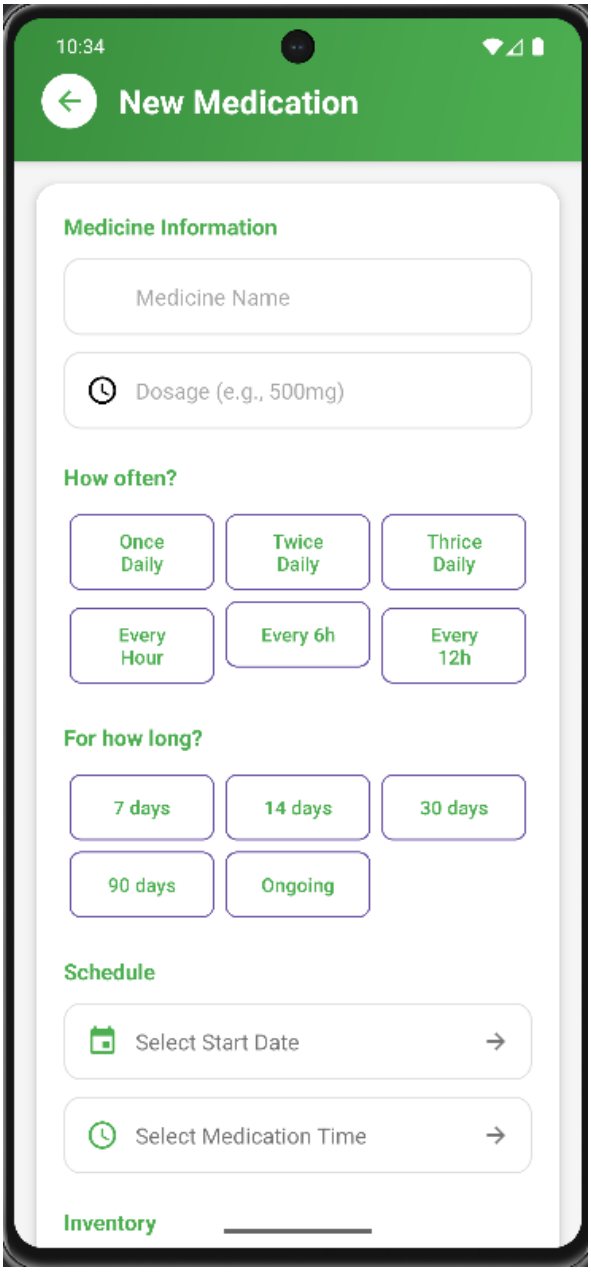
Profile Page

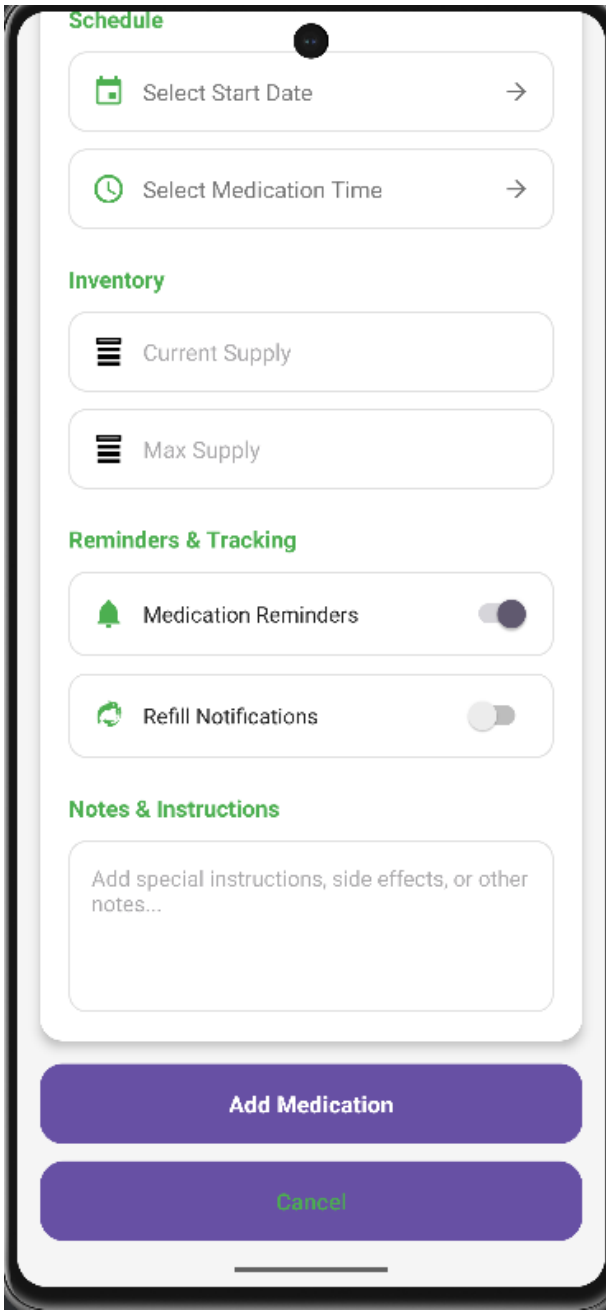


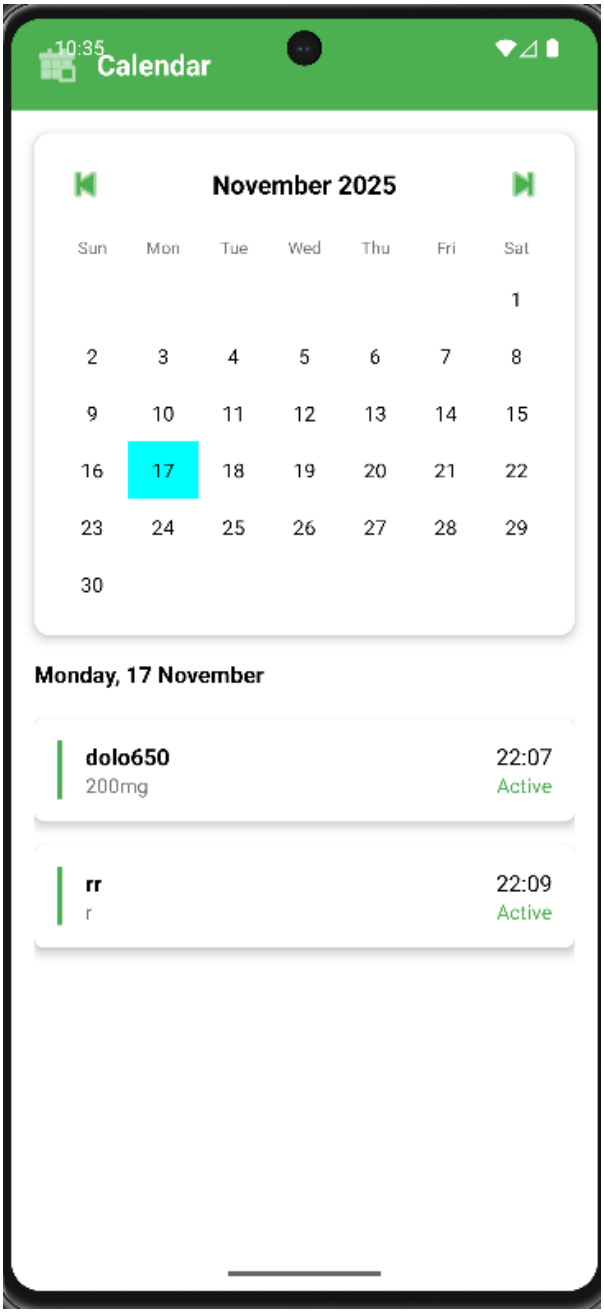
Edit Profile Page



Add Medication Page

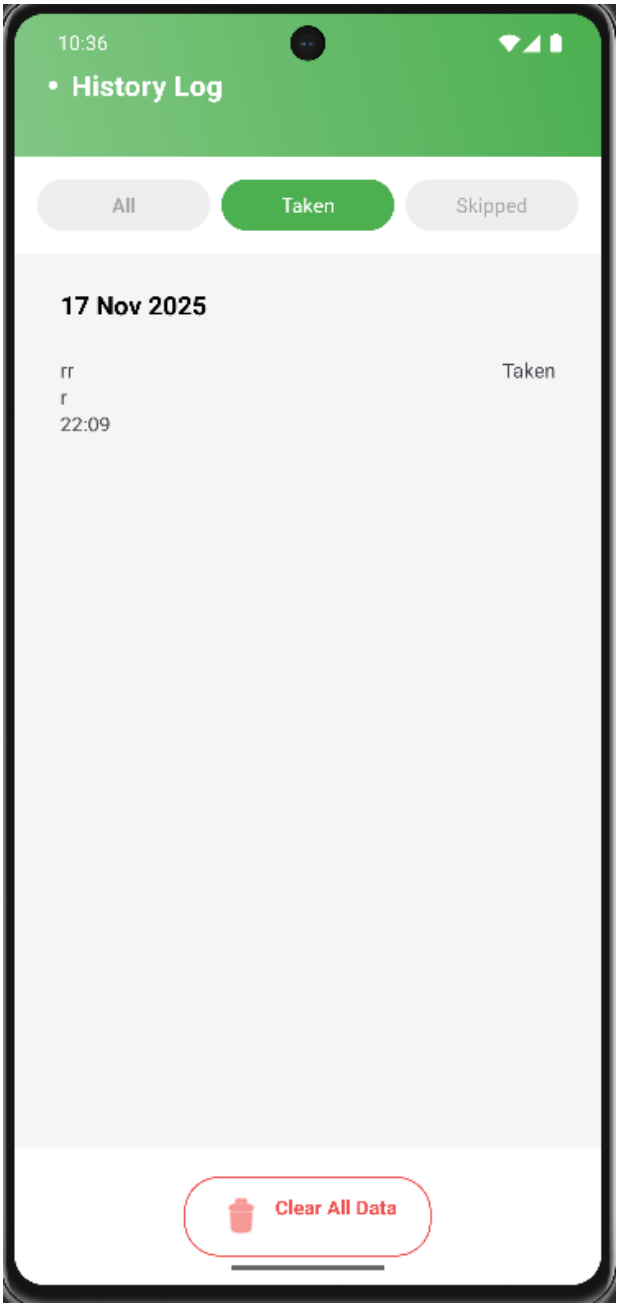




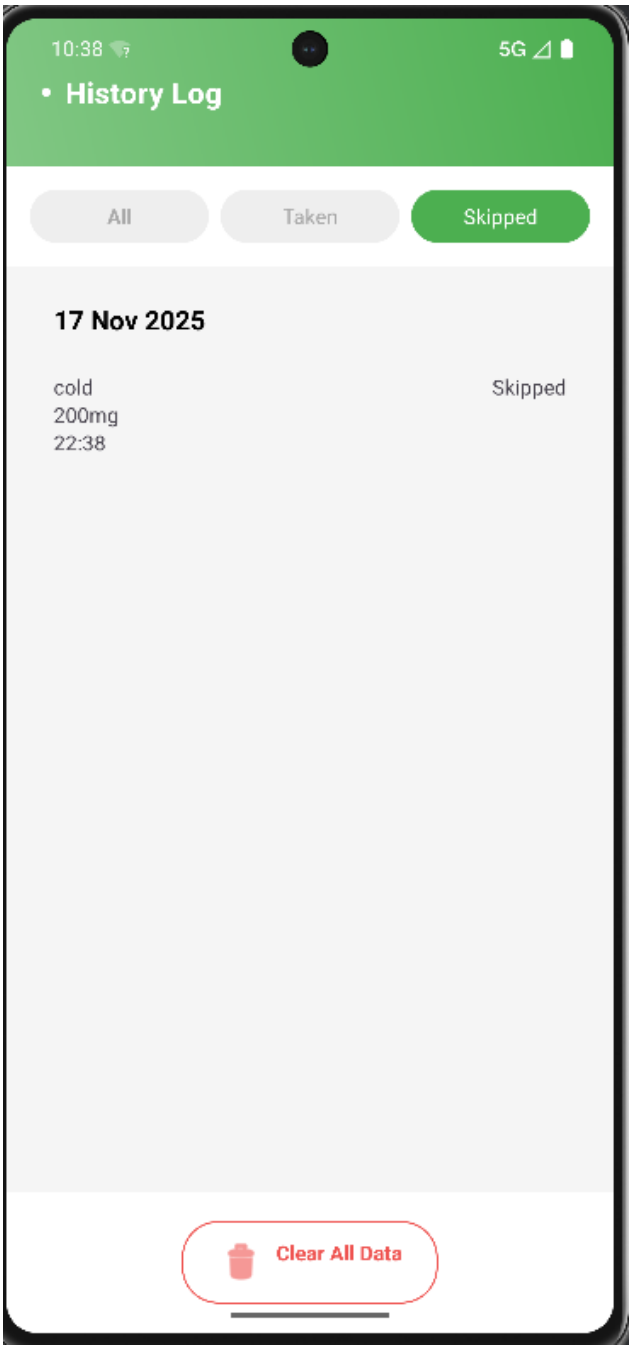


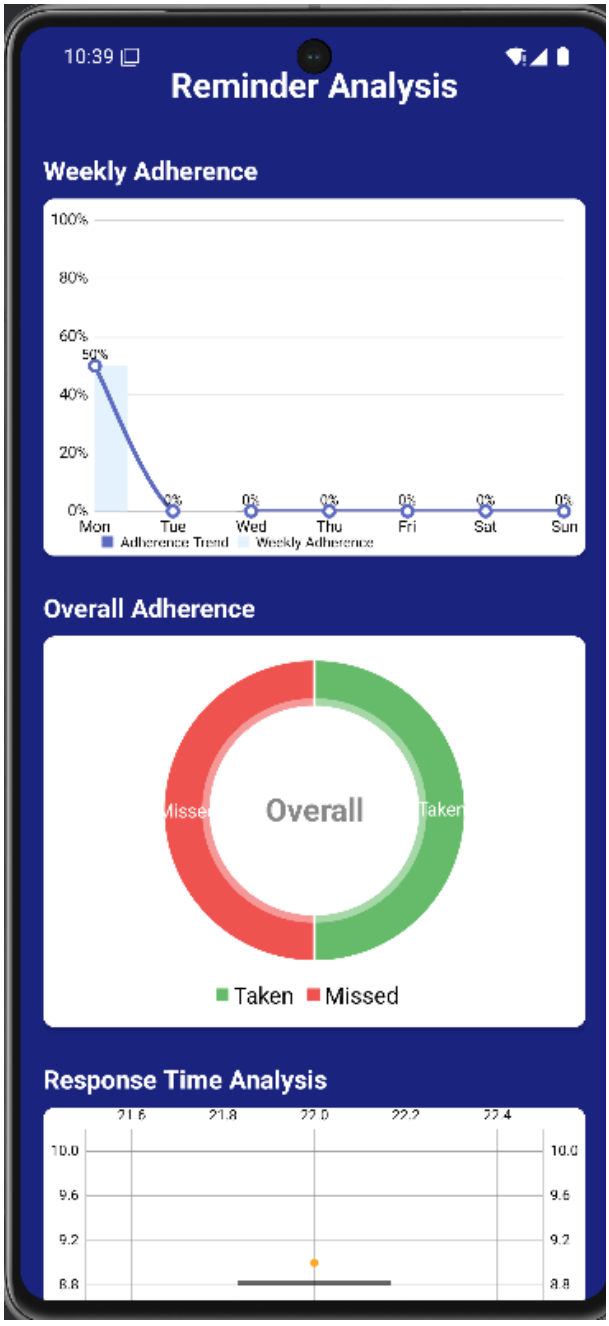
Calendar Page

History Page

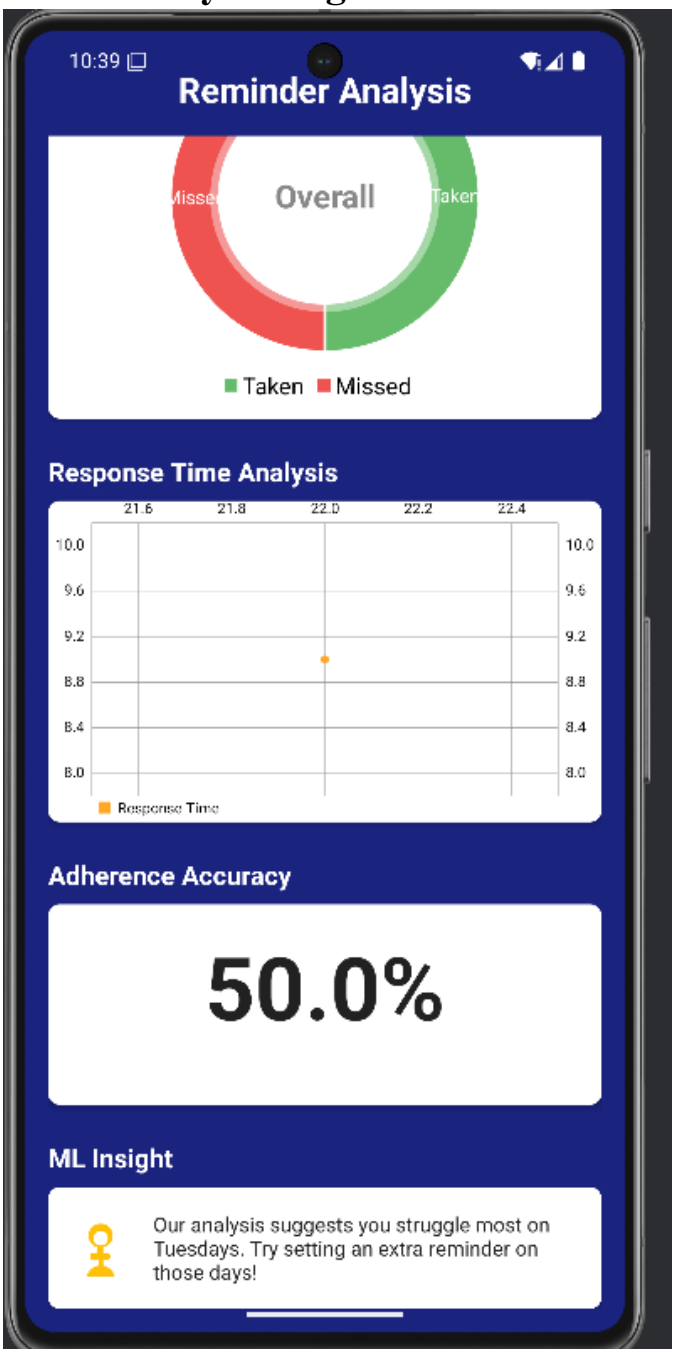


Chat Bot Page





View Analysis Page



Emergency Page