# CHAPTER 1

**INTRODUCTION**

## INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT

Mobile application development is the process of making software for smartphones and digital assistants, most commonly for Android and iOS. The software can be preinstalled on the device, downloaded from a mobile app store or accessed through a mobile web browser. The programming and markup languages used for this kind of software development include Java, Swift, Kotlin and so on.

Mobile app development is rapidly growing. From retail, telecommunications and e-commerce to insurance, healthcare and government, organizations across industries must meet user expectations for real-time, convenient ways to conduct transactions and access information. Today, mobile devices—and the mobile applications that unlock their value—are the most popular way for people and businesses to connect to the internet. To stay relevant, responsive and successful, organizations need to develop the mobile applications that their customers, partners and employee’s demand.

Yet mobile application development might seem daunting. Once you’ve selected the OS platform or platforms, you need to overcome the limitations of mobile devices and usher your app all the way past the potential hurdles of distribution. Fortunately, by following a few basic guidelines and best practices, you can streamline your application development journey.

There are four major development approaches when building mobile applications:

* + - Native Mobile Applications
    - Cross-Platform Native Mobile Applications
    - Hybrid Mobile Applications
    - Progressive Web Applications

### What is a Mobile Application?

A mobile application, also referred to as a mobile app or simply an app, is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch.

Apps were originally intended for productivity assistance such as email, calendar, and contact databases, but the public demand for apps caused rapid expansion into other areas such as mobile games, factory automation, GPS and location-based services, order-tracking, and ticket purchases, so that there are now millions of apps available. Apps are generally downloaded from application distribution platforms which are operated by the owner of the mobile operating system, such as the App Store (iOS) or Google Play Store. Some apps are free, and others have a price, with the profit being split between the application’s creators and the distribution platform. Mobile applications often stand in contrast to desktop applications which are designed to run on desktop computers, and web applications which run in mobile web browsers rather than directly on the mobile device.

### What is Mobile OS?

A mobile operating system (OS) is software that allows smartphones, tablet PCs (personal computers) and other devices to run applications and programs. A mobile OS typically starts up when a device powers on, presenting a screen with icons or tiles that present information and provide application access. Mobile operating systems also manage cellular and wireless network connectivity, as well as phone access.

### What is Mobile Application Development?

Mobile app development is the act or process by which a mobile app is developed for mobile devices, such as personal digital assistants, enterprise digital assistants or mobile phones. These applications can be pre-installed on phones during manufacturing platforms, or delivered as web applications using server-side or client-side processing (e.g., JavaScript) to provide an application- like experience within a Web browser.

To develop apps using the SDK, one mainly uses the Java or Kotlin programming language for developing the app and Extensible Markup Language (XML) files for describing data resources. By writing the code and creating a single app binary, you will have an app that can run on both phone and tablet. To help you develop your apps efficiently, Google offers a full Java Integrated Development Environment (IDE) called Android Studio, with advanced features for developing,debugging, and packaging Android apps. Using Android Studio, you can develop

apps on any available Android device, or create virtual devices that emulate any hardware configuration.

### Mobile Application Development Challenges

While the Android platform provide rich functionality for app development, there are still a number of challenges you need to address, such as:

* + - * Building for a multi-screen world
      * Getting performance right
      * Keeping your code and your users secure
      * Remaining compatible with older platform versions
      * Understanding the market and the user.

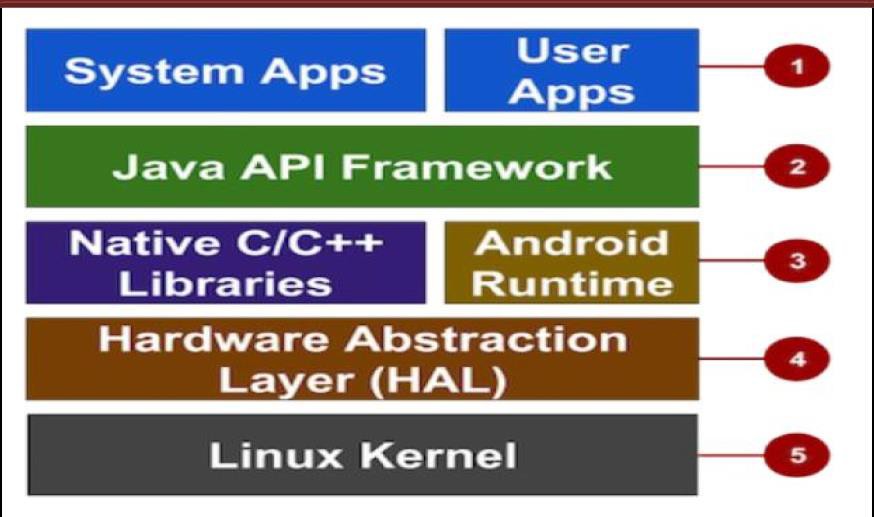
## INTRODUCTION TO ANDROID

Android is one of the most popular mobile device platforms. The Android platform allows developers to write managed code using Java to manage and control the Android device. Android Studio is a popular IDE developed by Google for developing applications that are targeted at the Android platform. Android Studio has replaced Eclipse as the IDE of choice for developing Android applications.

### What is Android?

Android is a software package and Linux based operating system for mobile devices such as tablet computers and smartphones. It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used.

### Android Development Architecture



**Figure 1.1 Android Development Architecture**

Apps: Your apps live at this level, along with core system apps for email, SMS messaging, calendars, Internet browsing, or contacts.

Java API Framework: All features of Android are available to developers through application programming interfaces (APIs) written in the Java language. You don’t need to know the details of all of the APIs to learn how to develop Android apps, but you can learn more about the following APIs, which are useful for creating apps:

View System used to build an app’s UI, including lists, buttons, and menus.

Resource Manager used to access non-code resources such as localized strings, graphics, and layout files.

Notification Manager used to display custom alerts in the status bar. Activity Manager that manages the life cycle of apps.

Libraries and Android Runtime: Each app runs in its own process and with its own instance of the Android Runtime, which enables multiple virtual machines on low-memory devices. Android also includes a set of core runtime libraries that provide most of the functionality of the Java

programming language, including some Java language features that the Java API framework uses. Many core Android system components and services are built from native code that requires native libraries written in C and C++.

## OVERVIEW OF THE PROJECT

The ‘Fitness App’ application has been promoting healthy lifestyle behavior, specifically healthy eating and weight control, has the potential to address our ultimate goal of enabling healthy lifestyle to prevent obesity and obesity related diseases. This app provides Pedometer for calculating the number of steps one takes in a day, BMI calculator to check one's body mass index, nutrition details of everything one consumes in a day, water tracker which helps in tracking the amount of water one consumes throughout the day and it lets you set a daily goal so that one is motivated to complete his/her goal for the day, exercise which has workout programs and clicking on each redirects you to YouTube. The major purpose of this project is to provide people with information about how to live a healthy lifestyle in this busy world with a single touch, avoid things which are bad for health, inculcating good habits by replacing the thing which is harmful for health with something which is easily accessible on a daily basis. ‘FIT ME-FITNESS APP’ application promotes healthy lifestyle behavior, it contains various diet plans which any individual can follow to lead a healthy lifestyle, with tips to replace unhealthy products to healthy products beneficial for health.

## PROBLEM STATEMENT

The fitness app incorporates features such as a Pedometer, BMI calculator, Nutrients, Water tracker, and Exercise, with the aim of providing users with a comprehensive tool to monitor and improve their overall health and fitness. This application implements a user-friendly interface that seamlessly integrates these features, ensuring accurate data collection and analysis while catering to users of varying fitness levels and goals. This application keeps the users motivated to maintain a healthy lifestyle with its easy understanding user interface.

# CHAPTER 2

**SYSTEM REQUIREMENTS SPECIFICATION**

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external behavior of the system.

## FUNCTIONAL REQUIREMENTS

Software under requirement definition is an abstract description of the services which the system should provide, and the constraints which the system must operate. It should only specify the external behavior of the system. Functional requirements in software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

## NON-FUNCTIONAL REQUIREMENTS

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and standards. Non-functional requirements often apply to the system as a whole.

Dependability: The dependability of a computer system is a property of the system that equates to its trustworthiness. Trustworthiness essentially means the degree of user confidence that the system will operate as they expect and that the system will not ‘fail’ in normal use.

Availability: The ability of the system to deliver services when requested. There is no error in the program while executing the program.

Reliability: The ability of the system to deliver services as specified. The program is compatible with all types of operating system without any failure.

Safety: The ability of the system to operate without catastrophic failure. This program is user friendly and it will never affect the system.

Security: The ability of the system to protect itself against accidental or deliberate intrusion.

## SOFTWARE REQUIREMENTS

The major constituent in the development of an android application is the sector of a compatible identification of resources. Selected software should be acceptable by the hardware as well as the platform we are working upon. It as well should be feasible for the system. Following are the description of the software specification.

1. OS: Windows 8 and above
2. Editor: Android Studio

## HARDWARE REQUIREMENTS

The section of hardware arrangement is an important undertaking related to the software part of the development process. It requires software to process and run correctly. Without the correct hardware most of the codes and programs would not have run properly. The process should be powerful enough to handle the entire operations of the applications and the appropriate hardware systems are needed to have sufficient capacity to accumulate the file and applications.

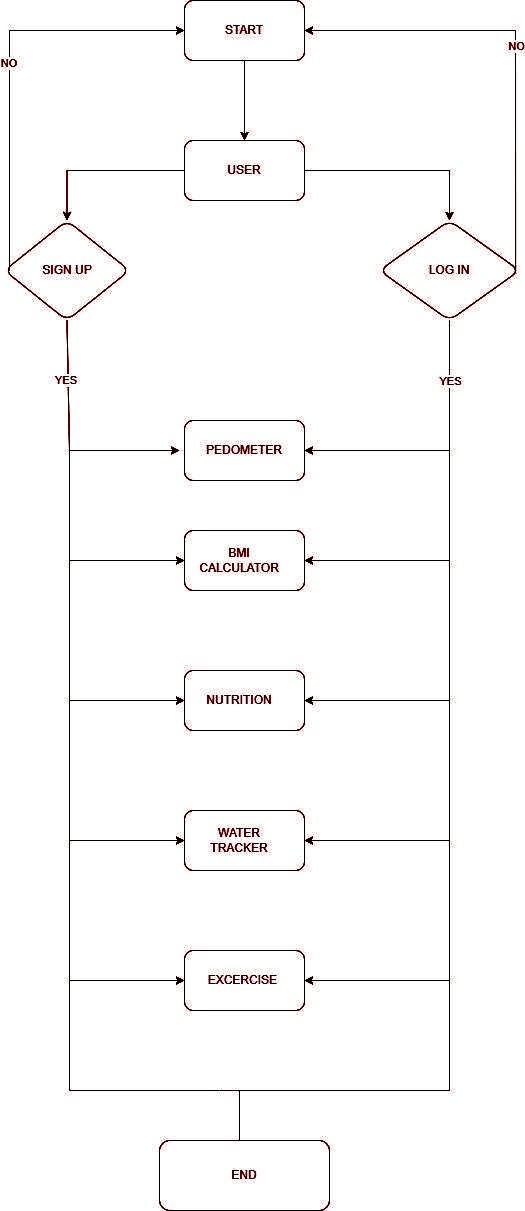
1. Pentium IV Processor, i3, i4, i5, i7
2. 8 GB RAM 3. 64-bit operating system

* The hardware requirements specified are the hardware components/capacity of the system in which the application is developed and deployed.
* The above software requirements are the necessary software required to develop the application and run the application using Android Studio.

# CHAPTER 3

**DESIGN**

## FLOW CHART



### Figure 3.1 Flowchart of the App

**Description of Flow Chart**

The description of the flow diagram is as follows:

Step 1: START

Step 2: The login page contains signup and signin pages. Step 3: Here we need to enter the email id and password.

Step 4: On clicking on signin we get redirected to the homepage if the login is successful. Step 5: Home page contains the main features which contains Pedometer,BMI,Nutrients,Water tracker, Exercise.

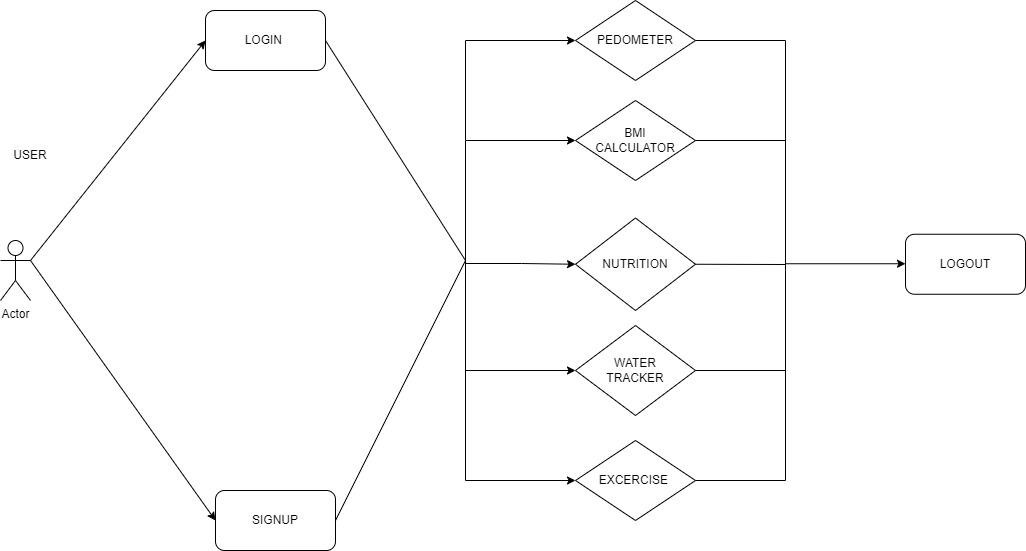
Step 6: : On clicking on the Pedometer we can count the number of steps. Step 7: On clicking on BMI it can calculate one's BMI.

Step 8: On clicking on Nutrition it shows the nutrient contents of Fruits, Vegetables, Carbs, Beverages, Dairy.

Step 9: On clicking Water it shows the place where one can track the amount of water consumed in a day.

Step 10: On clicking on Exercise it shows all the YouTube workout videos. Step 11: END

## USE CASE DIAGRAM



### Figure 3.2 Use case diagram of the App

The use case diagram of this fitness app demonstrates how users can interact with various features to track their physical activity, monitor their health metrics, manage their nutrition, and engage in guided exercises. The diagram provides a visual representation of the app's functionalities, highlighting the key interactions between the user and the app to support their fitness journey.

# CHAPTER 4

**IMPLEMENTATION**

## TECHNOLOGY USED:ANDROID STUDIO

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

1. Android app modules
2. Library modules
3. Google App Engine modules

By default, Android Studio displays your project files in the Android project view. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under Gradle Scripts and each app module contains the following folders:

* manifests: Contains the AndroidManifest.xml file.
* java: Contains the Java source code files, including JUnit test code.
* res: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images

### FEATURES OF ANDROID STUDIO:

* Code and iterate faster than ever.
* Fast and feature-rich emulator.
* Help to Build Up App for All Devices.
* Support KOTLIN (Official language used for android).
* Color previews.
* Addition of New Activity as a Code Template.
* Lint tools to catch performance, usability, version compatibility, and other problems.
* A flexible Gradle-based build system

## Android studio has 3 types of code completion:

1. **Basic completion:**

Displays basic suggestions for variables, types, methods, expressions, and so on. If you call basic completion twice in a row, you see more results including private members and non imported static members.

Windows/Mac: Ctrl+Space

## Smart completion:

Displays relevant options based on the context. Smart completion is aware of the expected type and data flows. If you call Smart completion twice in a row, you see more results including chains.

Windows/Mac:Ctrl+Shift+Space

## Statement Completion:

Completes the current statement for you, adding missing parentheses, brackets, braces, formatting etc.

Windows: Ctrl+Shift+Enter Mac: Shift+Command+Enter

## SOURCE CODE

### AndroidManifest.xml

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

[<manifest xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) package="com.example.fitme">

<uses-permission android:name="android.permission.ACTIVITY\_RECOGNITION" android:required="true" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-feature

android:name="android.hardware.sensor.stepdetector" android:required="true" />

<uses-feature android:name="android.hardware.sensor.accelerometer" android:required="true" />

<uses-feature android:name="android.hardware.sensor.stepcounter" android:required="true" />

<application

android:allowBackup="true" android:icon="@drawable/fitness\_icon" android:label="@string/app\_name" android:roundIcon="@drawable/fitness\_icon" android:supportsRtl="true" android:theme="@style/Theme.fitme">

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

<activity

android:name=".excerciseActivity" android:exported="false" />

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

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

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

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

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

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

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

<activity android:name=".MainActivity" android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

### activity\_bmi.xml

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

<androidx.constraintlayout.widget.ConstraintLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

[xmlns:app="http://schemas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) [xmlns:tools="http://schemas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent" tools:context=".BmiActivity">

<TextView android:id="@+id/Bmi\_title" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_marginTop="36dp" android:text="@string/YourBmiText" android:textColor="@color/black" android:textSize="36sp" android:textStyle="bold" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

<TextView android:id="@+id/bmi\_value" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_marginTop="88dp" android:text="@string/bmiValue" android:textColor="#DA1212" android:textSize="100sp" android:textStyle="bold" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/Bmi\_title" />

<TextView android:id="@+id/bmi\_category" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginTop="70dp" android:textColor="@color/black" android:textSize="26sp"

app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/bmi\_value" />

<TextView android:id="@+id/bmi\_tips" android:layout\_width="350dp"

android:layout\_height="wrap\_content" android:layout\_marginTop="60dp" android:text="@string/bmiTips" android:textAlignment="center" android:textColor="#DA1212" android:textSize="18sp"

app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/bmi\_category" />

<Button

android:id="@+id/calculate\_again\_btn" android:layout\_width="330dp" android:layout\_height="wrap\_content" android:layout\_marginBottom="24dp" android:background="@drawable/button\_design" android:text="Calculate" android:textColor="#FFFFFF" android:textSize="18sp" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

### activity\_main.xml

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

<androidx.constraintlayout.widget.ConstraintLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

[xmlns:app="http://schemas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) [xmlns:tools="http://schemas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:background="#6DC5BC" tools:context=".MainActivity">

<androidx.constraintlayout.widget.Guideline android:id="@+id/glCenterHorizontal"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:orientation="horizontal" app:layout\_constraintGuide\_percent="0.5" />

<TextView android:id="@+id/tvAlmostThere" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="28dp" android:layout\_marginBottom="24dp" android:clickable="true" android:text="Almost there!" android:textColor="#FFFFFF" android:textSize="30sp" android:textStyle="bold"

app:layout\_constraintBottom\_toTopOf="@+id/tvExcited" app:layout\_constraintStart\_toStartOf="parent" tools:ignore="TouchTargetSizeCheck" />

<TextView android:id="@+id/tvExcited" android:layout\_width="0dp"

android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginEnd="32dp" android:layout\_marginBottom="68dp" android:text="Enter the following Entities" android:textColor="#FFFFFF" android:textSize="30sp"

android:textStyle="bold" app:layout\_constraintBottom\_toTopOf="@+id/tilEmail" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.0" app:layout\_constraintStart\_toStartOf="parent" />

<TextView android:id="@+id/textView2" android:layout\_width="335dp" android:layout\_height="25dp" android:textColor="#E46B6B"

app:layout\_constraintBottom\_toTopOf="@+id/tilEmail" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.542" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/tvExcited" app:layout\_constraintVertical\_bias="0.44" />

<com.google.android.material.textfield.TextInputLayout android:id="@+id/tilEmail" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginEnd="32dp" android:layout\_marginBottom="8dp" android:textColorHint="#FFFFFF" app:hintTextColor="#FFFFFF"

app:layout\_constraintBottom\_toTopOf="@+id/glCenterHorizontal" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="@+id/glCenterHorizontal" app:layout\_constraintVertical\_bias="0.906">

<com.google.android.material.textfield.TextInputEditText android:id="@+id/editTextTextEmailAddress" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:hint="Email" android:inputType="textEmailAddress" android:maxLines="1"

android:textColor="#FFFFFF" android:textColorHint="#FFFFFF" tools:ignore="TouchTargetSizeCheck" />

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

<com.google.android.material.textfield.TextInputLayout android:id="@+id/tilPassword" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginTop="8dp" android:layout\_marginEnd="32dp" android:textColorHint="#FFFFFF" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.0" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/tilEmail">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/editTextTextPassword" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:hint="Password" android:inputType="textPassword" android:maxLines="1" android:textColorHint="#FFFFFF" tools:ignore="TouchTargetSizeCheck" />

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

<CheckBox android:id="@+id/checkBox" android:layout\_width="322dp" android:layout\_height="36dp" android:layout\_marginStart="24dp" android:layout\_marginEnd="24dp" android:text="I Agree" android:textColor="#2196F3"

app:layout\_constraintBottom\_toTopOf="@+id/signup\_btn" app:layout\_constraintEnd\_toEndOf="@+id/tilPassword" app:layout\_constraintStart\_toStartOf="@+id/tilPassword" app:layout\_constraintTop\_toBottomOf="@+id/tilPassword" tools:ignore="TouchTargetSizeCheck" />

<Button

android:id="@+id/signup\_btn" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginStart="32dp" android:layout\_marginTop="60dp"

android:layout\_marginEnd="32dp" android:background="#FFFEFE" android:text="Sign up" android:textAllCaps="false" android:textColor="#6DC5BC" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.0" app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/tilPassword" tools:ignore="TextContrastCheck" />

<TextView android:id="@+id/tvHaveAnAccount" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="8dp" android:layout\_marginTop="16dp" android:text="Already have an account?" android:textColor="#FFFFFF" android:textSize="14sp"

app:layout\_constraintEnd\_toStartOf="@+id/tvSignIn" app:layout\_constraintHorizontal\_bias="0.5" app:layout\_constraintHorizontal\_chainStyle="packed" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/signup\_btn" />

<TextView android:id="@+id/tvSignIn"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp" android:layout\_marginEnd="8dp" android:text="Sign in." android:textColor="#FFFFFF" android:textStyle="bold" android:clickable="true" android:onClick="signinn" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toEndOf="@+id/tvHaveAnAccount" app:layout\_constraintTop\_toTopOf="@+id/tvHaveAnAccount" />

</androidx.constraintlayout.widget.ConstraintLayout>

### activity\_pedometer.xml

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

[<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) [xmlns:app="http://schemas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) [xmlns:tools="http://schemas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:background="#6DC5BC" android:orientation="vertical" tools:context=".pedometerActivity">

<androidx.cardview.widget.CardView android:layout\_width="300dp" android:layout\_height="300dp" android:layout\_gravity="center" android:innerRadius="0dp" android:shape="ring"

app:cardBackgroundColor="#FFFFFF" app:cardCornerRadius="110dp" app:cardElevation="6dp" app:cardUseCompatPadding="true">

<LinearLayout android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="center" android:gravity="center" android:orientation="vertical">

<ImageView android:layout\_width="100dp" android:layout\_height="100dp" android:src="@drawable/walk" />

<TextView android:id="@+id/stepText"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="0" android:textColor="#3D6CFC" android:textSize="30dp" />

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Steps Count" />

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView android:layout\_width="match\_parent" android:layout\_height="80dp" app:cardCornerRadius="10dp"

app:cardElevation="10dp" app:cardUseCompatPadding="true">

<TextView android:id="@+id/timeText"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="center" android:drawablePadding="10dp" android:gravity="center" android:text="0" android:textColor="#3D6CFC" android:textSize="16sp" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView android:layout\_width="match\_parent" android:layout\_height="80dp" app:cardCornerRadius="10dp" app:cardElevation="10dp" app:cardUseCompatPadding="true">

<TextView android:id="@+id/distanceText"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="center" android:drawablePadding="10dp" android:gravity="center" android:text="0" android:textColor="#3D6CFC" android:textSize="16sp" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView android:layout\_width="match\_parent" android:layout\_height="80dp" app:cardCornerRadius="10dp" app:cardElevation="10dp" app:cardUseCompatPadding="true">

<TextView android:id="@+id/orientationText" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="center" android:drawablePadding="10dp" android:gravity="center" android:text="East" android:textColor="#3D6CFC" android:textSize="16sp" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView android:layout\_width="match\_parent" android:layout\_height="80dp"

app:cardCornerRadius="10dp" app:cardElevation="10dp" app:cardUseCompatPadding="true">

<TextView android:id="@+id/speedText" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="center" android:drawablePadding="10dp" android:gravity="center" android:text="0" android:textColor="#3D6CFC" android:textSize="16sp" />

</androidx.cardview.widget.CardView>

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:orientation="horizontal" android:paddingStart="10dp" android:paddingRight="10dp">

<Button

android:id="@+id/startButton" android:layout\_width="50dp" android:layout\_height="40dp" android:layout\_marginRight="10dp" android:layout\_weight="1"

android:background="#FFFFFF" android:text="@string/start" android:textColor="#3D6CFC" android:textSize="20sp" />

<Button

android:id="@+id/resetButton" android:layout\_width="50dp" android:layout\_height="40dp" android:layout\_weight="1" android:background="#FFFFFF" android:text="RESET" android:textColor="#3D6CFC" android:textSize="20sp" />

</LinearLayout>

</LinearLayout>

### activity\_watertrack.xml

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

[<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) [xmlns:app="http://schemas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) [xmlns:tools="http://schemas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:background="#6DC5BC" android:orientation="vertical" android:weightSum="5" tools:context=".watertrackActivity">

<LinearLayout

android:layout\_width="match\_parent" android:layout\_height="0dp" android:layout\_weight="3" android:orientation="vertical">

<Button

android:id="@+id/set\_goal" android:layout\_width="51dp" android:layout\_height="38dp" android:layout\_gravity="right" android:layout\_marginTop="10dp" android:layout\_marginRight="10dp" android:layout\_marginBottom="30dp" android:background="@android:color/transparent"

android:drawableEnd="@drawable/ic\_settings\_black\_24dp" android:onClick="showPopup" android:stateListAnimator="@null" />

<ImageView android:id="@+id/img\_drop" android:layout\_width="200dp" android:layout\_height="200dp" android:layout\_gravity="center" android:layout\_marginBottom="30dp" app:srcCompat="@drawable/drop0" />

<TextView android:id="@+id/sum\_goal" android:layout\_width="170dp" android:layout\_height="50dp"

android:layout\_gravity="center" android:textAlignment="center" android:textColor="#03A9F4" android:textSize="18sp" android:textStyle="bold" />

<TextView android:id="@+id/sum\_percentage" android:layout\_width="170dp" android:layout\_height="90dp" android:layout\_gravity="center" android:textAlignment="center" android:textColor="#03A9F4" android:textSize="40sp" android:textStyle="bold" />

</LinearLayout>

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="0dp" android:layout\_weight="1" android:gravity="center" android:orientation="horizontal">

<Button

android:id="@+id/add" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginRight="20dp" android:background="@drawable/button\_shape\_round"

android:text="@string/button\_text250" android:textColor="#FFFFFF" android:textSize="18sp" android:textStyle="bold" />

<Button

android:id="@+id/select" android:layout\_width="64dp" android:layout\_height="wrap\_content" android:background="@android:color/transparent"

android:drawableStart="@drawable/ic\_menu\_black\_24dp" android:textColor="#FFFFFF"

android:textSize="10sp" android:textStyle="bold" />

</LinearLayout>

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="0dp" android:layout\_weight="1" android:gravity="center" android:orientation="vertical">

<TextView android:id="@+id/textBox2" android:layout\_width="346dp" android:layout\_height="90dp" android:gravity="center" android:textAlignment="center" android:textAllCaps="false"

android:textColor="#03A9F4" android:textSize="18sp" android:textStyle="bold" />

</LinearLayout>

</LinearLayout>

### BmiActivity.java

package com.example.fitme;

import androidx.appcompat.app.AppCompatActivity; import android.os.Bundle;

import android.view.View; import android.widget.Button; import android.widget.TextView; import com.example.fitme.R;

public class BmiActivity extends AppCompatActivity { TextView bmiValue, bmiCategory,bmiTips;

String category;

String bmiValOutput;

Button calculateAgainBtn; String[] bmiTipsArray; @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_bmi);

bmiValue = findViewById(R.id.bmi\_value); bmiCategory = findViewById(R.id.bmi\_category); bmiTips = findViewById(R.id.bmi\_tips);

bmiTipsArray = getResources().getStringArray(R.array.tips\_array); calculateAgainBtn = findViewById(R.id.calculate\_again\_btn); bmiValOutput = getIntent().getStringExtra("bmiVal");

bmiValue.setText(bmiValOutput); findCategory();

categoryTips();

calculateAgainBtn.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) { onBackPressed();

}

});

}

private void categoryTips() {

double result = Double.parseDouble(bmiValOutput); if(result < 15){

bmiTips.setText(bmiTipsArray[0]);

}

else if(result >= 15 && result <= 16){ bmiTips.setText(bmiTipsArray[0]);

}

else if(result >= 16 && result <= 18.5){ bmiTips.setText(bmiTipsArray[1]);

}

else if(result >= 18.5 && result <= 25){ bmiTips.setText(bmiTipsArray[2]);

}

else if(result >= 25 && result <= 30){ bmiTips.setText(bmiTipsArray[3]);

}

else if(result >=30 && result <= 35){ bmiTips.setText(bmiTipsArray[4]);

}

else if(result >= 35 && result <= 50){ bmiTips.setText(bmiTipsArray[4]);

}

else

bmiTips.setText(bmiTipsArray[4]);

}

private void findCategory() {

double result = Double.parseDouble(bmiValOutput); if(result < 15){

category = "Very Severely Underweight"; bmiCategory.setText(category);

}

else if(result >= 15 && result <= 16){ category = "Severely Underweight"; bmiCategory.setText(category);

}

else if(result >= 16 && result <= 18.5){ category = "Underweight"; bmiCategory.setText(category);

}

else if(result >= 18.5 && result <= 25){ category = "Normal (Healthy weight)"; bmiCategory.setText(category);

}

else if(result >= 25 && result <= 30){ category = "Overweight"; bmiCategory.setText(category);

}

else if(result >=30 && result <= 35){ category = "Moderately Obese";

bmiCategory.setText(category);

}

else if(result >= 35 && result <= 50){ bmiCategory.setText(category); category = "Severely Obese";

}

else

category = "Very Severely Obese"; bmiCategory.setText(category);

}

}

### MainActivity.java

package com.example.fitme;

import android.content.DialogInterface; import android.content.Intent;

import androidx.activity.result.contract.ActivityResultContracts; import androidx.annotation.NonNull;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity; import android.os.Bundle;

import android.util.Log; import android.view.View; import android.widget.Button;

import android.widget.CheckBox; import android.widget.EditText; import android.widget.TextView; import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task; import com.google.firebase.auth.AuthResult; import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.auth.FirebaseUser;

public class MainActivity extends AppCompatActivity { private static final String TAG = "MainActivity"; public FirebaseAuth mAuth;

Button signUpButton;

EditText signUpEmailTextInput; EditText signUpPasswordTextInput; CheckBox agreementCheckBox; TextView errorView;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

mAuth = FirebaseAuth.getInstance();

signUpEmailTextInput = findViewById(R.id.editTextTextEmailAddress); signUpPasswordTextInput = findViewById(R.id.editTextTextPassword); signUpButton = findViewById(R.id.signup\_btn);

agreementCheckBox = findViewById(R.id.checkBox); errorView = findViewById(R.id.textView2); signUpButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (signUpEmailTextInput.getText().toString().contentEquals("")) { errorView.setText("Email cannot be empty");

} else if (signUpPasswordTextInput.getText().toString().contentEquals("")) { errorView.setText("Password cannot be empty");

} else if (!agreementCheckBox.isChecked()) {

errorView.setText("Please agree to terms and Condition");

} else { mAuth.createUserWithEmailAndPassword(signUpEmailTextInput.getText().toString(),

signUpPasswordTextInput.getText().toString()).addOnCompleteListener(MainActivity.this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) { if (task.isSuccessful()) {

// Sign in success, update UI with the signed-in user's information Log.d(TAG, "createUserWithEmail:success");

FirebaseUser user = mAuth.getCurrentUser(); try { if (user != null)

user.sendEmailVerification().addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) { if (task.isSuccessful()) {

Log.d(TAG, "Email sent.");

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder( MainActivity.this); alertDialogBuilder.setTitle("Please Verify Your EmailID");

alertDialogBuilder

.setMessage("A verification Email Is Sent To Your Registered EmailID, please click on the link and Sign in again!")

.setCancelable(false).setPositiveButton("SignIn",newDialogInterface.OnClickListener()

{ public void onClick(DialogInterface dialog, int id) {

Intent signInIntent = new Intent(MainActivity.this, SignInActivity.class); startActivity(signInIntent);

}

});

AlertDialog alertDialog = alertDialogBuilder.create(); alertDialog.show();

}

}

});

} catch(Exception e){ errorView.setText(e.getMessage());

}

} else {

Log.w(TAG, "createUserWithEmail:failure", task.getException());

Toast.makeText(MainActivity.this, "Authentication failed.", Toast.LENGTH\_SHORT).show();

if (task.getException() != null) { errorView.setText(task.getException().getMessage());

}

});

}

}

}

}

});

}

public void signinn(View view) {

Intent in= new Intent(MainActivity.this,SignInActivity.class); startActivity(in);

}

}

### pedometerActivity.java

package com.example.fitme;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.content.ContextCompat; import android.content.Context;

import android.content.DialogInterface; import android.content.SharedPreferences; import android.hardware.Sensor;

import android.hardware.SensorEvent;

import android.hardware.SensorEventListener; import android.hardware.SensorManager; import android.os.Bundle;

import android.os.Handler; import android.os.SystemClock;

import android.preference.PreferenceManager; import android.view.View;

import android.widget.Button; import android.widget.TextView;

public class pedometerActivity extends AppCompatActivity implements SensorEventListener { private SensorManager sensorManager;

private Sensor stepDetectorSensor; private Sensor accelerometer; private Sensor magnetometer; private float[] accelValues;

private float[] magnetValues; private int stepCount = 0; private long stepTimestamp = 0; private long startTime = 0;

long timeInMilliseconds = 0; long elapsedTime = 0;

long updatedTime = 0; private double distance = 0;

private TextView dayRecordText;

private TextView stepText; private TextView timeText; private TextView orientationText; private TextView distanceText; private TextView achievedText; private TextView speedText;

private boolean active = false; //Used to checked if the counter is running private Handler handler = new Handler(); //Used to update the time in the UI private SharedPreferences sharedPreferences;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_pedometer);

sensorManager = (SensorManager) getSystemService(Context.SENSOR\_SERVICE); stepDetectorSensor = sensorManager.getDefaultSensor(Sensor.TYPE\_STEP\_DETECTOR); accelerometer = sensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER); magnetometer = sensorManager.getDefaultSensor(Sensor.TYPE\_MAGNETIC\_FIELD);

if (stepDetectorSensor == null) showErrorDialog();

sharedPreferences = PreferenceManager.getDefaultSharedPreferences(pedometerActivity.this);

stepText = findViewById(R.id.stepText); timeText = findViewById(R.id.timeText); speedText = findViewById(R.id.speedText); distanceText = findViewById(R.id.distanceText);

orientationText = findViewById(R.id.orientationText); setViewDefaultValues();

final Button startButton = findViewById(R.id.startButton); if (startButton != null) {

startButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) { if (!active) {

startButton.setText(R.string.pause); startButton.setBackgroundColor(ContextCompat.getColor(pedometerActivity.this,

R.color.darkGray));

sensorManager.registerListener(pedometerActivity.this, stepDetectorSensor, SensorManager.SENSOR\_DELAY\_NORMAL);

sensorManager.registerListener(pedometerActivity.this, accelerometer, SensorManager.SENSOR\_DELAY\_NORMAL);

sensorManager.registerListener(pedometerActivity.this, magnetometer, SensorManager.SENSOR\_DELAY\_NORMAL);

startTime = SystemClock.uptimeMillis(); handler.postDelayed(timerRunnable, 0); active = true;

} else {

startButton.setText(R.string.start); startButton.setBackgroundColor(ContextCompat.getColor(pedometerActivity.this,

R.color.lightGray));

sensorManager.unregisterListener(pedometerActivity.this, stepDetectorSensor); sensorManager.unregisterListener(pedometerActivity.this, accelerometer); sensorManager.unregisterListener(pedometerActivity.this, magnetometer); elapsedTime += timeInMilliseconds; handler.removeCallbacks(timerRunnable);

active = false;

}

}

});

}

Button resetButton =findViewById(R.id.resetButton);

resetButton.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) { stepCount = 0;

distance = 0;

elapsedTime = 0; setViewDefaultValues();

}

});

}

@Override

public void onSensorChanged(SensorEvent event) { switch (event.sensor.getType()) {

case (Sensor.TYPE\_ACCELEROMETER): accelValues = event.values;

break;

case (Sensor.TYPE\_MAGNETIC\_FIELD): magnetValues = event.values;

break;

case (Sensor.TYPE\_STEP\_DETECTOR): countSteps(event.values[0]); calculateSpeed(event.timestamp); break;

}

if (accelValues != null && magnetValues != null) { float rotation[] = new float[9];

float orientation[] = new float[3];

if (SensorManager.getRotationMatrix(rotation, null, accelValues, magnetValues)) { SensorManager.getOrientation(rotation, orientation);

float azimuthDegree = (float) (Math.toDegrees(orientation[0]) + 360) % 360;

float orientationDegree = Math.round(azimuthDegree); getOrientation(orientationDegree);

}

}

}

@Override

public void onAccuracyChanged(Sensor sensor, int i) {

}

private void setViewDefaultValues() { stepText.setText(String.format(getResources().getString(R.string.steps), 0));

timeText.setText(String.format(getResources().getString(R.string.time), "0:00:00"));

speedText.setText(String.format(getResources().getString(R.string.speed), 0));

distanceText.setText(String.format(getResources().getString(R.string.distance), "0")); orientationText.setText(String.format(getResources().getString(R.string.orientation), ""));

}

private void countSteps(float step) { stepCount += (int) step;

stepText.setText(String.format(getResources().getString(R.string.steps), stepCount)); distance = stepCount \* 0.8; //Average step length in an average adult

String distanceString = String.format("%.2f", distance); distanceText.setText(String.format(getResources().getString(R.string.distance),

distanceString));

}

private void calculateSpeed(long eventTimeStamp) {

long timestampDifference = eventTimeStamp - stepTimestamp; stepTimestamp = eventTimeStamp;

double stepTime = timestampDifference / 1000000000.0; int speed = (int) (60 / stepTime);

speedText.setText(String.format(getResources().getString(R.string.speed), speed));

}

private void getOrientation(float orientationDegree) { String compassOrientation;

if (orientationDegree >= 0 && orientationDegree < 90) { compassOrientation = "North";

} else if (orientationDegree >= 90 && orientationDegree < 180) { compassOrientation = "East";

} else if (orientationDegree >= 180 && orientationDegree < 270) { compassOrientation = "South";

} else {

compassOrientation = "West";

}

orientationText.setText(String.format(getResources().getString(R.string.orientation),

compassOrientation));

}

private Runnable timerRunnable = new Runnable() { @Override

public void run() {

timeInMilliseconds = SystemClock.uptimeMillis() - startTime; updatedTime = elapsedTime + timeInMilliseconds;

int seconds = (int) (updatedTime / 1000); int minutes = seconds / 60;

int hours = minutes / 60; seconds = seconds % 60; minutes = minutes % 60;

String timeString = String.format("%d:%s:%s", hours, String.format("%02d", minutes), String.format("%02d", seconds));

timeText.setText(String.format(getResources().getString(R.string.time), timeString)); handler.postDelayed(this, 0);

}

};

private void showErrorDialog() {

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(pedometerActivity.this); alertDialogBuilder.setMessage("Necessary step sensors not available!"); alertDialogBuilder.setPositiveButton("Exit", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) { pedometerActivity.this.finish();

}

});

AlertDialog alertDialog = alertDialogBuilder.create(); alertDialog.show();

}

}

### watertrackActivity.java

package com.example.fitme; import android.app.AlarmManager; import android.app.PendingIntent; import android.content.Intent;

import android.content.SharedPreferences; import android.graphics.Color;

import android.os.Bundle;

import android.view.ContextThemeWrapper; import android.view.MenuItem;

import android.view.View;

import android.view.animation.Animation; import android.view.animation.AnimationUtils; import android.widget.Button;

import android.widget.ImageView; import android.widget.PopupMenu;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity; import java.text.SimpleDateFormat;

import java.util.Calendar; import java.util.Locale;

public class watertrackActivity extends AppCompatActivity implements PopupMenu.OnMenuItemClickListener, Dialog.DialogListener, NumberPickerDialog.DialogListener {

private int sum = 0;

private int defaultGoal = 3700; private int defaultPick = 250; private static TextView mainText;

private static TextView mainPercentage; private static TextView mainGoal; private static ImageView imgView;

int[] images = {R.drawable.drop0, R.drawable.drop1, R.drawable.drop2, R.drawable.bmi, R.drawable.drop4, R.drawable.drop5, R.drawable.drop6, R.drawable.drop7, R.drawable.drop8, R.drawable.drop9, R.drawable.drop10};

public static final String sharedPref = "sharedPref"; private String date;

private String dateToCompare; private String text;

private String percentageText;

private String lastColor = "#FFFFFF"; private Animation animation;

private SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy.MM.dd", Locale.getDefault());

private AlarmManager alarmManager; private PendingIntent pi;

private Button mainButton; public void setSum(int sum) {

this.sum = sum;

}

public int getDefaultGoal() { return defaultGoal;

}

public int getSum() { return sum;

}

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_watertrack); variableEq();

initializeAlarmManager(); loadData(); updateViews(); checkForNextDay();

}

public void variableEq() {

mainText = findViewById(R.id.sum\_goal); mainPercentage = findViewById(R.id.sum\_percentage); mainGoal = findViewById(R.id.textBox2);

imgView = findViewById(R.id.img\_drop);

animation = AnimationUtils.loadAnimation(watertrackActivity.this, R.anim.animation\_drop);

mainButton = findViewById(R.id.add); mainButton.setOnClickListener(Global\_OnClickListener); mainButton.setText(String.valueOf(defaultPick));

findViewById(R.id.select).setOnClickListener(Global\_OnClickListener);

}

public void initializeAlarmManager() {

Intent intent = new Intent(this, Receiver.class);

pi = PendingIntent.getBroadcast(this, 0, intent, 0);

alarmManager = (AlarmManager) this.getSystemService(ALARM\_SERVICE); alarmManager.set(AlarmManager.RTC\_WAKEUP, System.currentTimeMillis() +

AlarmManager.INTERVAL\_HOUR, pi);

}

public String goalReached() {

return "Today's goal has been reached!" + " You drank " + getSum() + " ml!";

}

public void over() {

if (getSum() > getDefaultGoal()) { lastColor = "#B82441";

mainText.setTextColor(Color.parseColor(lastColor)); mainPercentage.setTextColor(Color.parseColor(lastColor));

}

}

public void refresh() {

mainText.setText(getSum() + "/" + getDefaultGoal() + "ml");

}

public void reset() { setSum(sum = 0); lastColor = "#FFFFFF";

mainText.setTextColor(Color.parseColor(lastColor)); mainPercentage.setTextColor(Color.parseColor(lastColor)); mainText.setText(getSum() + "/" + getDefaultGoal() + "ml"); mainPercentage.setText((getSum() \* 100) / getDefaultGoal() + "%"); mainGoal.setText("");

imgView.setImageResource(setImg(sum, false));

}

public int setImg(int sum, boolean reached) { if (!reached) {

if (sum == 0) return images[0];

if (sum <= (defaultGoal \* 0.1)) return images[1];

if (sum <= (defaultGoal \* 0.2)) return images[2];

if (sum <= (defaultGoal \* 0.3)) return images[3];

if (sum <= (defaultGoal \* 0.4)) return images[4];

if (sum <= (defaultGoal \* 0.5)) return images[5];

if (sum <= (defaultGoal \* 0.6)) return images[6];

if (sum <= (defaultGoal \* 0.7)) return images[7];

if (sum <= (defaultGoal \* 0.8)) return images[8];

if (sum <= (defaultGoal \* 0.9)) return images[9];

if (sum == defaultGoal) return images[10];

}

return images[10];

}

public void setTxt() {

over();

mainText.setText(getSum() + "/" + getDefaultGoal() + "ml"); mainPercentage.setText((getSum() \* 100) / getDefaultGoal() + "%");

}

public void setImage(boolean reached) { imgView.setImageResource(setImg(sum, reached));

}

public void goalReach() { mainGoal.setText(goalReached());

}

public void buttonAction(int amount) { setSum(sum += amount);

setTxt(); setImage(false); saveData();

if (sum >= defaultGoal) { goalReach(); setImage(true); saveData();

}

imgView.startAnimation(animation);

}

final View.OnClickListener Global\_OnClickListener = new View.OnClickListener() { public void onClick(final View v) {

switch (v.getId()) { case R.id.add:

buttonAction(defaultPick); break;

case R.id.select: openDialogNumberPicker(); break;

}

}

};

@Override

public boolean onMenuItemClick(MenuItem item) { switch (item.getItemId()) {

case R.id.item1: openDialogSettings(); return true;

case R.id.item2: reset();

return true; default:

return false;

}

}

public void saveData() {

SharedPreferences sharedPreferences = getSharedPreferences(sharedPref, MODE\_PRIVATE);

SharedPreferences.Editor editor = sharedPreferences.edit(); date = dateFormat.format(Calendar.getInstance().getTime());

editor.putString("color", lastColor); editor.putString("text", mainText.getText().toString());

editor.putString("percentage", mainPercentage.getText().toString()); editor.putInt("sum", getSum());

editor.putInt("defaultGoal", getDefaultGoal());

editor.putString("time", date); editor.putInt("pick", defaultPick); editor.apply();

}

public void loadData() {

SharedPreferences sharedPreferences = getSharedPreferences(sharedPref, MODE\_PRIVATE);

lastColor = sharedPreferences.getString("color", lastColor); percentageText = sharedPreferences.getString("percentage", ""); text = sharedPreferences.getString("text", "");

date = sharedPreferences.getString("time", ""); sum = sharedPreferences.getInt("sum", sum);

defaultGoal = sharedPreferences.getInt("defaultGoal", defaultGoal); defaultPick = sharedPreferences.getInt("pick", defaultPick); imgView.setImageResource(setImg(sum, false));

dateToCompare = dateFormat.format(Calendar.getInstance().getTime()); mainButton.setText(String.valueOf(defaultPick));

}

public void checkForNextDay() {

if (date.compareTo(dateToCompare) < 0) reset();

}

public void updateViews() { mainText.setText(text); mainPercentage.setText(percentageText);

mainText.setTextColor(Color.parseColor(lastColor)); mainPercentage.setTextColor(Color.parseColor(lastColor));

}

public void showPopup(View v) {

ContextThemeWrapper ctw = new ContextThemeWrapper(this, R.style.ItemTheme);

PopupMenu popup = new PopupMenu(ctw, v); popup.setOnMenuItemClickListener(this); popup.inflate(R.menu.pop\_menu); popup.show();

}

public void showInfo(View v) { openDialogInfo();

}

public void openDialogSettings() { Dialog dialog = new Dialog();

dialog.show(getSupportFragmentManager(), "dialog");

}

public void openDialogInfo() { InfoDialog dialog = new InfoDialog();

dialog.show(getSupportFragmentManager(), "dialog");

}

public void openDialogNumberPicker() { NumberPickerDialog dialog = new NumberPickerDialog(); dialog.show(getSupportFragmentManager(), "dialog");

}

@Override

public void applyText(String newGoal) { defaultGoal = Integer.parseInt(newGoal); reset();

refresh(); saveData();

}

@Override

public void applySumButton(int newSelection) { defaultPick = newSelection;

mainButton.setText(String.valueOf(defaultPick)); refresh();

saveData();

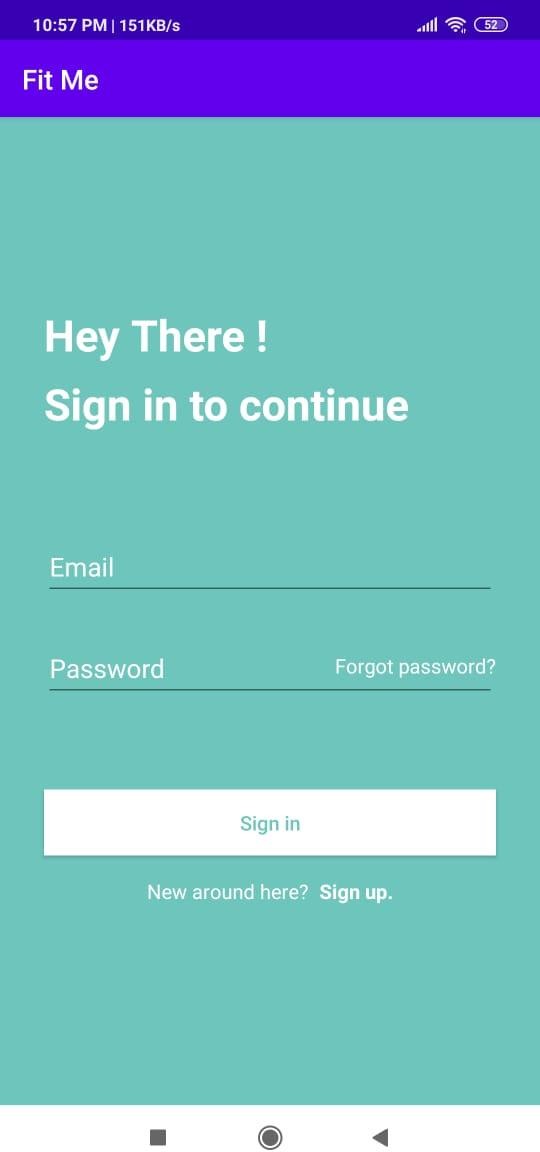
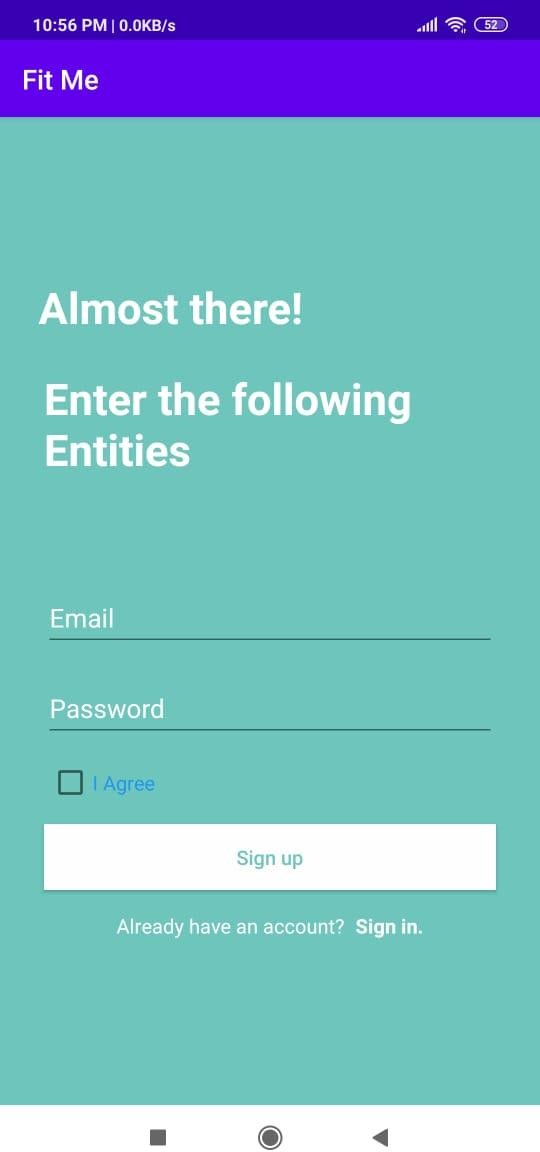
}

}

# CHAPTER 5

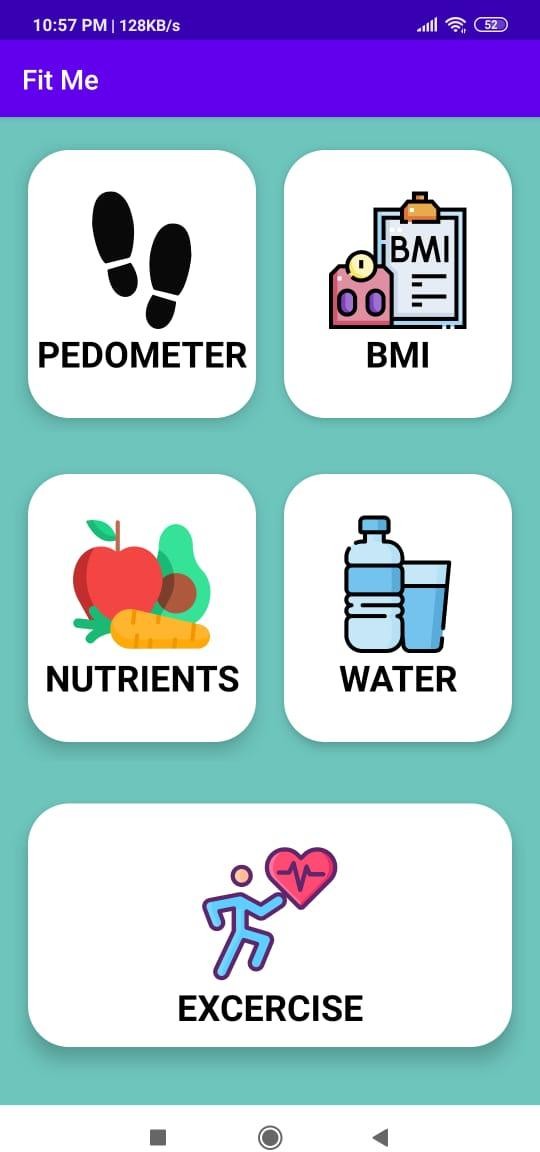
**SNAPSHOTS AND RESULTS**

This section contains some snapshots from the application, showing the various options available and a brief overview of how the application looks.



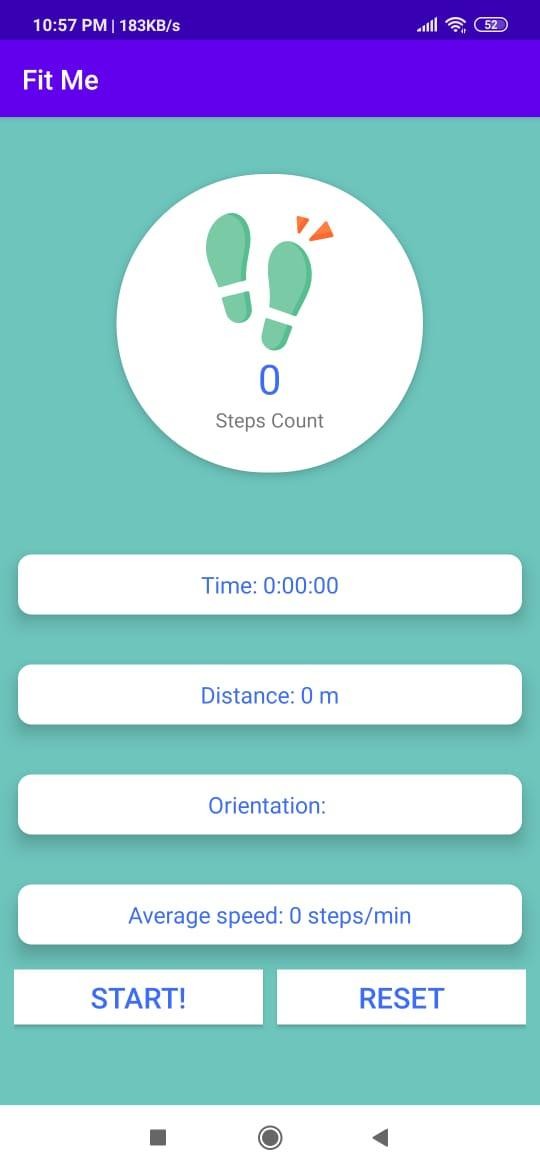
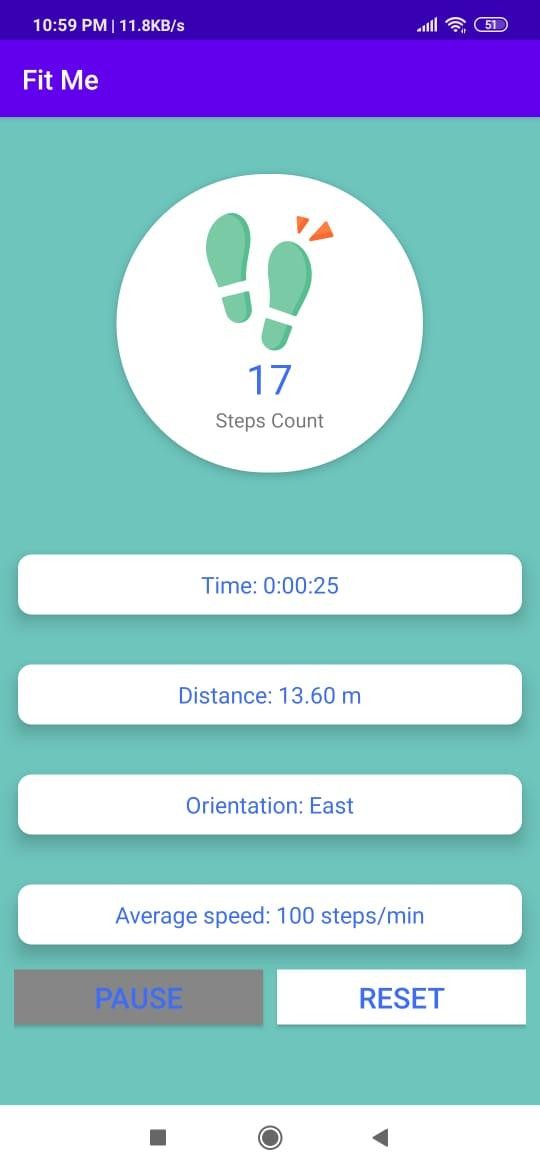
### Figure 5.1 SIGN UP and SIGN IN Page

This page asks one to enter the email id and password in order to access the features of the “Fit Me” application.



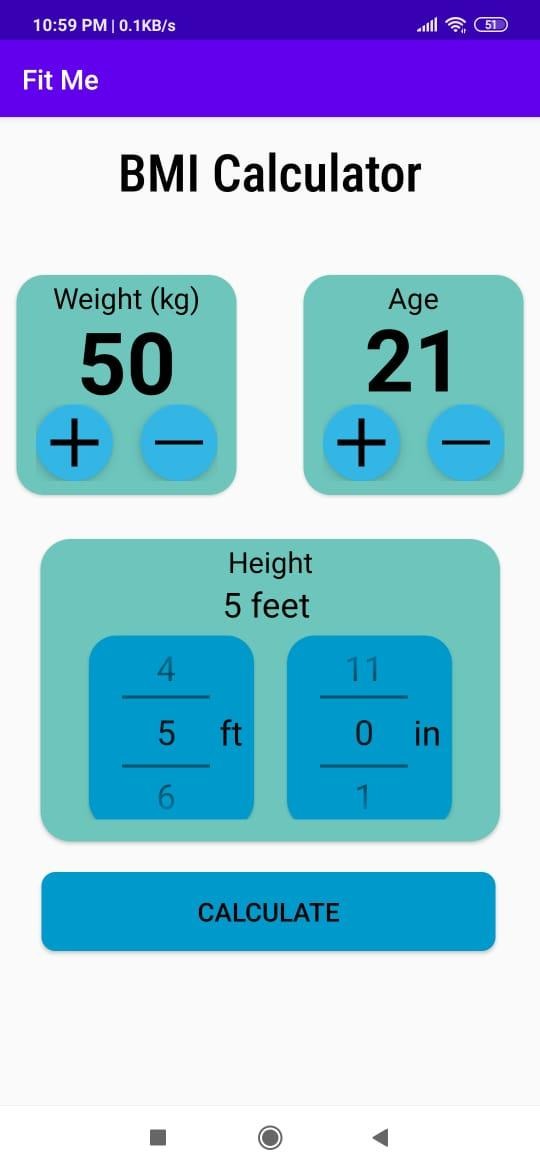
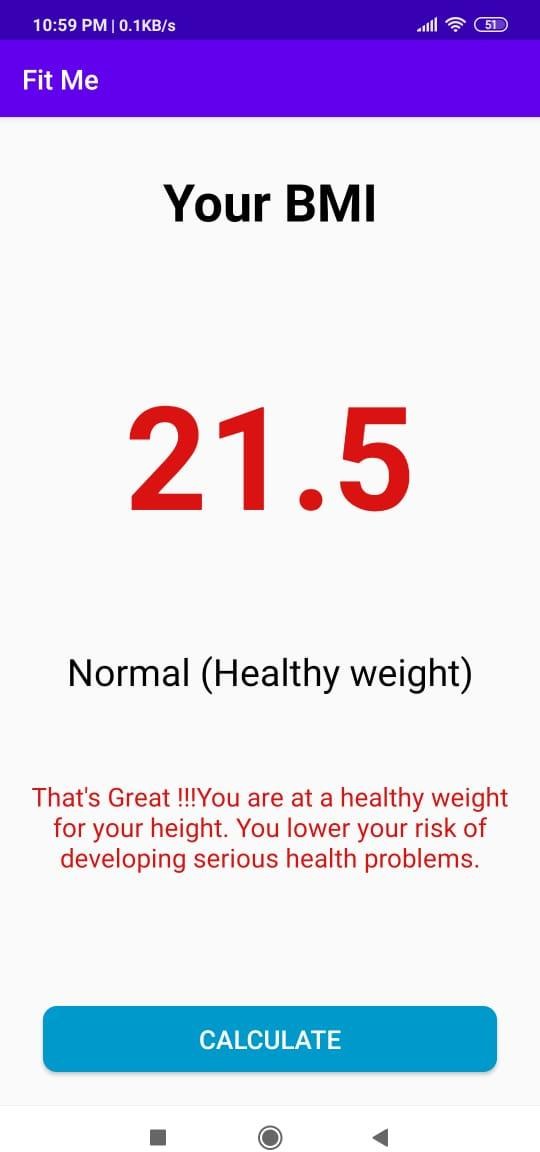
### Figure 5.2 Home Page

Fitness application incorporating features such as a Pedometer, BMI calculator, Nutrients, Water tracker, and Exercise would provide an overview of how users interact with the app's functionality.

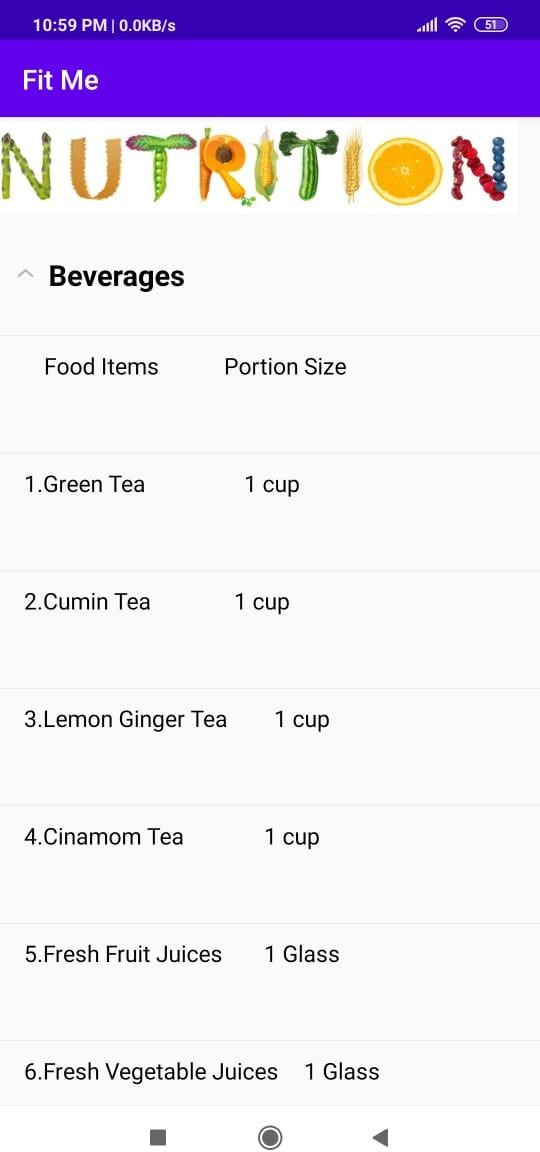
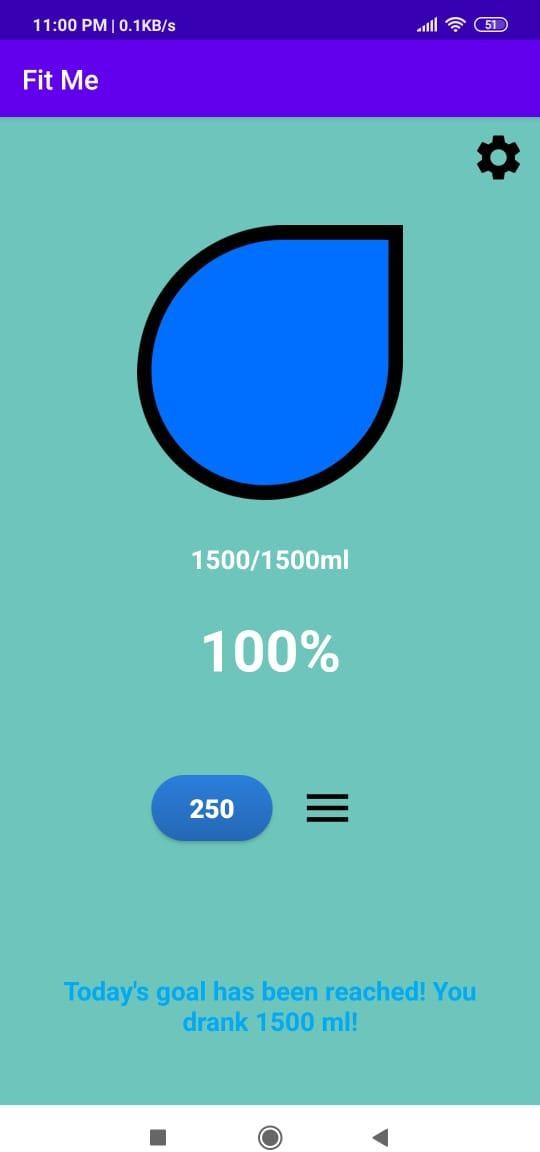
### Figure 5.3 Pedometer Page Figure 5.4 Pedometer in action Page

The Pedometer feature would enable the user to track their daily steps and set goals for physical activity. The user would be able to start and stop the pedometer tracker, view step counts, and receive the duration of the walk, direction in which the walk is towards, total distance traveled and the average speed.

**Figure 5.5 BMI Calculator Page Figure 5.6 BMI Calculated Page**

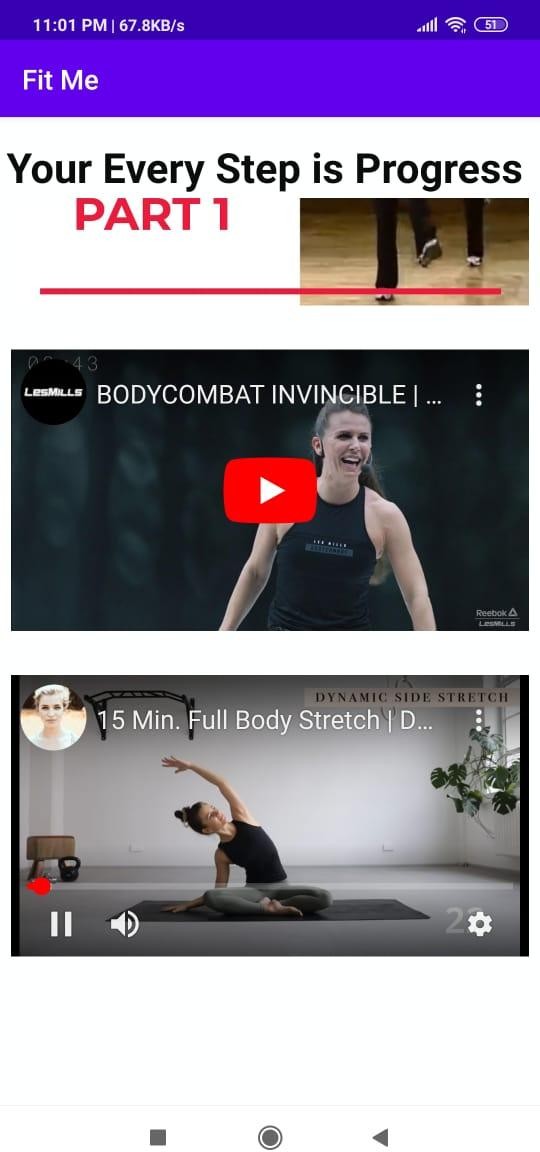
The BMI calculator feature would allow users to input their height and weight information, and the app would calculate and display their body mass index. Users could also track their BMI over time, helping them monitor their progress towards their health and fitness goals.

### Figure 5.7 Nutrition Page Figure 5.8 Water Tracker Page

The Nutrients feature in Figure 5.7 would help a user follow a proper healthy diet by providing information regarding the nutrition values of some of the food items.

The Water tracker feature in figure 5.8 would help users monitor their daily water intake. Users could set goals for the amount of water they want to consume, and the app provides graphical user interface and visual displays help them stay hydrated throughout the day.

**Figure 5.9 Exercise Page**

Exercise features would offer users a variety of workout options and routines. The app could provide instructional videos for different types of exercises. Users would be able to choose workouts based on their fitness level and preferences, and the app might even offer personalized recommendations based on their goals and previous activities.

# CHAPTER 6

**CONCLUSION AND FUTURE ENHANCEMENT**

## CONCLUSION

The “Fitness App '' application is very useful and important for people to make sure that their health is going well or not through their smartphones and tablets. The main aim of this project is to facilitate people in fitness, exercising and improving their knowledge regarding nutrition, number of steps taken, body mass index, amount of water consumed in a day. It helps the user to live a healthy lifestyle in this busy world,where there is no time to rush to the gym. This app provides structured home-workout for different age groups of people and which can be done in less than an hour. This app helps the user to get all information related to fitness on a single platform with a finger smooth UI.

## FUTURE ENHANCEMENT

* + - In the future, the same project can be enhanced in such a way that we can add notes (syncing with google) for daily achieving fitness tasks.
    - A fitness related blog section can be included so that individuals can get vast knowledge about fitness and health at one platform.
    - Adding a notification or sound for the water tracker section so that one can get daily reminders to drink water.
    - Adding features to customized diet plans according to goal and personal preferences.

## Reference Books

**BIBLIOGRAPHY**

1. Google Developer Training, "Android Developer Fundamentals Course – Concept Reference”, Google Developer Training Team, 2017.
2. Erik Hellman, “Android Programming – Pushing the Limits”, 1st Edition, Wiley India Pvt Ltd, 2014.

Websites

* 1. [www.google.com](http://www.google.com/)
  2. [www.wikipedia.org](http://www.wikipedia.org/)
  3. [www.geeksforgeeks.org](http://www.geeksforgeeks.org/)