**CODE**

**1)activity\_main.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="@color/white"

tools:context=".MainActivity">

<ImageView

android:id="@+id/header"

android:layout\_width="match\_parent"

android:layout\_height="200dp"

android:layout\_margin="0dp"

android:src="@drawable/header"

android:scaleType="centerCrop"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<Chronometer

android:id="@+id/chronometer"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="130dp"

app:layout\_constraintTop\_toTopOf="parent"

android:paddingTop="25dp"

android:textSize="70sp"

android:background="@drawable/bg"

android:textAlignment="center"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent" />

<ToggleButton

android:id="@+id/Toggle"

android:layout\_width="150dp"

android:layout\_height="150dp"

android:background="@drawable/start\_pause\_btn\_bg"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

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

<Button

android:id="@+id/reset\_btn"

android:layout\_width="90dp"

android:layout\_height="90dp"

android:background="@drawable/reset\_button"

app:layout\_constraintBottom\_toBottomOf="@+id/Toggle"

app:layout\_constraintStart\_toEndOf="@+id/Toggle" />

</androidx.constraintlayout.widget.ConstraintLayout>

**2)AndroidManifest.xml:**

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

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools">

<application

android:allowBackup="true"

android:dataExtractionRules="@xml/data\_extraction\_rules"

android:fullBackupContent="@xml/backup\_rules"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme"

tools:targetApi="31">

<activity

android:name=".MainActivity"

android:exported="true">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

</manifest>

**3)start\_pause\_btn\_bg.xml:**

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

<selector xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:state\_checked="false"

android:drawable="@drawable/start\_button"

/>

<item

android:state\_checked="true"

android:drawable="@drawable/pause" />

</selector>

**4)colors.xml:**

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

<resources>

<color name="colorPrimary">#52dbfe</color>

<color name="colorPrimaryDark">#369ee3</color>

<color name="colorAccent">#D81B60</color>

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

</resources>

**5)themes.xml:**

<resources>

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

<item name="colorPrimary">@color/colorPrimary</item>

<item name="colorPrimaryDark">@color/colorPrimaryDark</item>

<item name="colorAccent">@color/colorAccent</item>

<item name="android:statusBarColor">@color/colorPrimary</item>

<item name="android:theme">@style/AppTheme</item>

</style>

</resources>

**6)MainActivity.java:**

package com.example.timer;

import android.os.Bundle;

import android.os.SystemClock;

import android.view.View;

import android.widget.Button;

import android.widget.Chronometer;

import android.widget.CompoundButton;

import android.widget.ToggleButton;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

private Chronometer chronometer;

private long PauseOffSet = 0;

private boolean isPlaying = false;

private ToggleButton toggleButton;

private Button reset\_btn;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//Let's select our element

chronometer = findViewById(R.id.chronometer);

toggleButton = findViewById(R.id.Toggle);

reset\_btn = findViewById(R.id.reset\_btn);

// now we need to customize our toggle button

toggleButton.setText(null);

toggleButton.setTextOn(null);

toggleButton.setTextOff(null);

//the event listener will automaticly generated by android studio

toggleButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(CompoundButton compoundButton, boolean b) {

//now let's code our stopwatch

//b is the state of our button, it will be true if we pressed the button otherwise it will be false

if(b){

chronometer.setBase(SystemClock.elapsedRealtime()- PauseOffSet);

chronometer.start();

isPlaying = true;

}else{

chronometer.stop();

PauseOffSet = SystemClock.elapsedRealtime()- chronometer.getBase();

isPlaying = false;

}

}

});

//now we need to add the reset function of our button

// we should check if the chronometre is actually workin to be able to reset otherwise we can't

reset\_btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if(isPlaying){

chronometer.setBase(SystemClock.elapsedRealtime());

PauseOffSet = 0;

chronometer.start();

isPlaying = true;

//now everything should work let's run our app and see it

}

}

});

}

}