**PRACTICAL: 3**

**AIM:** Create an application that will display Toast (Message) on specific interval of time

**THEORY:**

An Android Toast is a small message displayed on the screen, similar to a tool tip or other similar popup notification. A Toast is displayed on top of the main content of an activity, and only remains visible for a short time period.

In Android, Chronometer is a class that implements a simple timer. Chronometer is a subclass of TextView. This class helps us to add a timer in our app.

You can give Timer start time in the elapsedRealTime() timebase and it start counting from that. If we don’t give base time then it will use the time at which time we call start() method. By default a chronometer displays the current timer value in the form of MM:SS or H:MM:SS. We can set our own format into an arbitrary string.

OnChronometerTickListener(Chronometer.OnChronometerTickListener listener): This is a listener event which is automatically called when the chronometer changes.

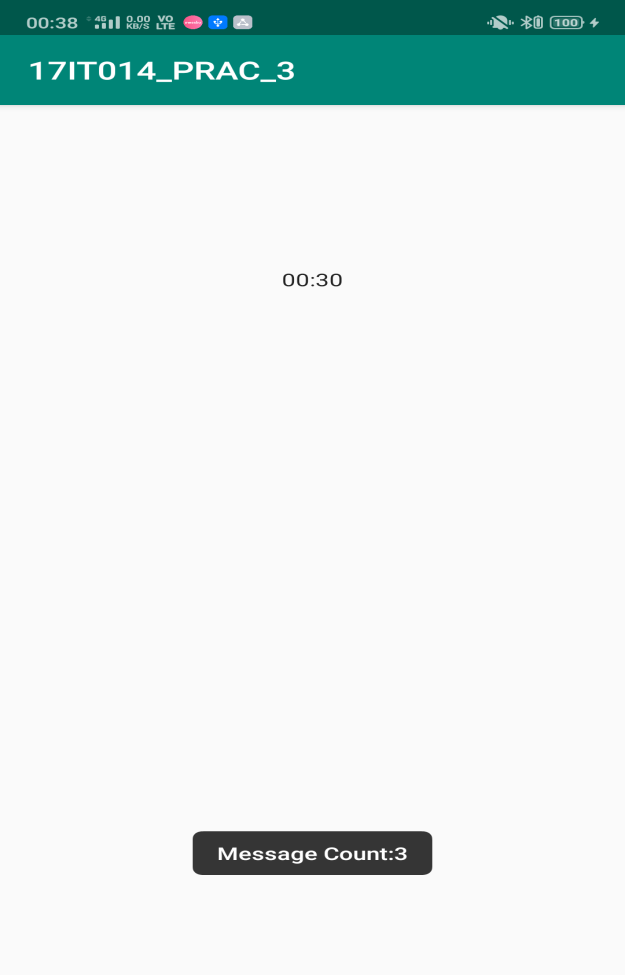
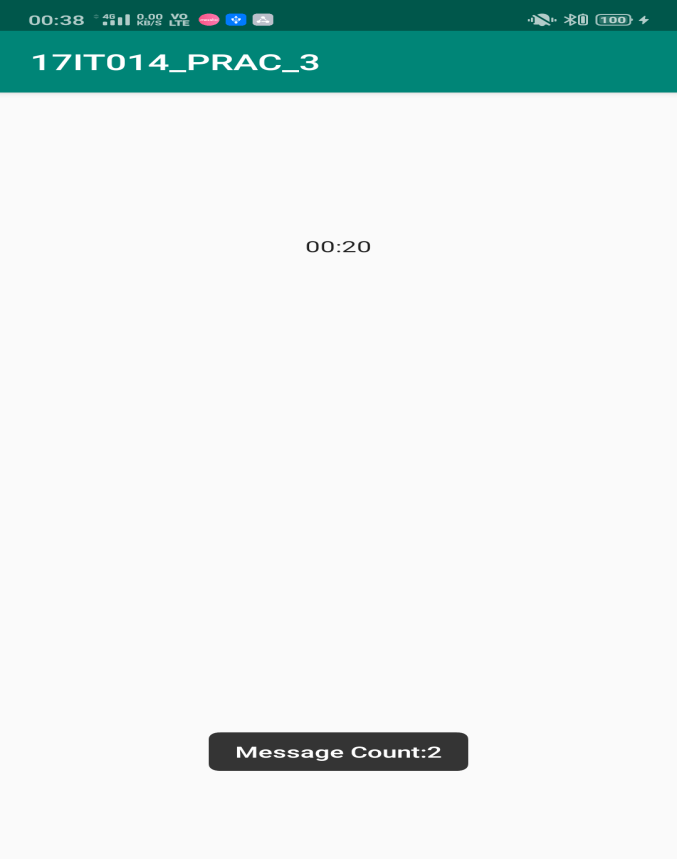
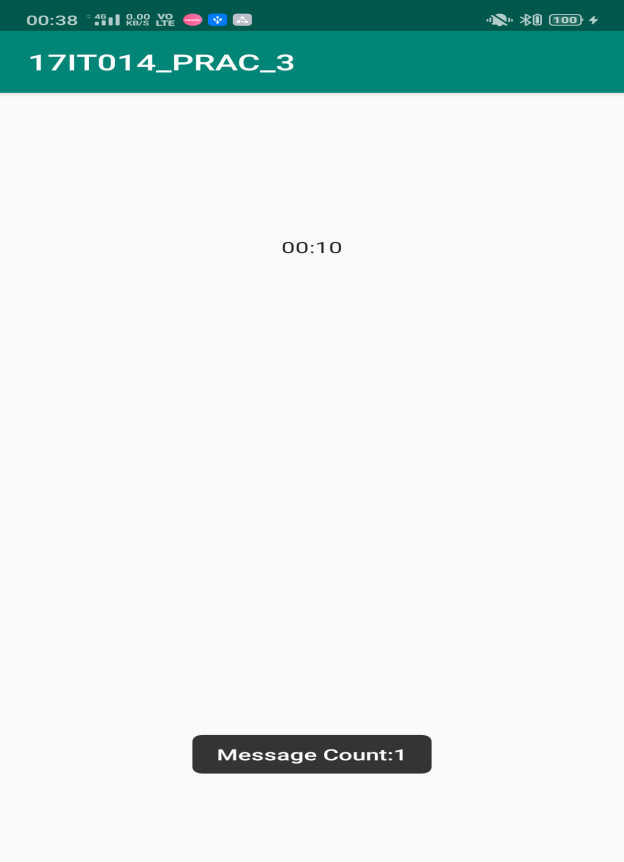
**CODE:**

|  |
| --- |
| **MainActivity.java**  package com.example.test\_pr3;  import androidx.appcompat.app.AppCompatActivity;  import android.os.Bundle;  import android.widget.Chronometer;  import android.widget.Toast;  import java.util.Timer;  public class MainActivity extends AppCompatActivity {  private Chronometer timer;  int i=-1;  int duration=10;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  timer=(Chronometer)findViewById(R.id.chronometer);  timer.start();  timer.setOnChronometerTickListener(new Chronometer.OnChronometerTickListener() {  @Override  public void onChronometerTick(Chronometer chronometer) {  i++;  if (i >= duration) {  Toast.makeText(MainActivity.this, "Message Count:" + (i / 10), Toast.LENGTH\_LONG).show();  duration = duration + 10;  }  }  });  }  }  **activity\_main.xml**  <?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"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">  <Chronometer  android:id="@+id/chronometer"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentTop="true"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="130dp" />  </RelativeLayout>  **AndroidManifest.xml**  <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.test\_pr3">  <application  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="17IT014\_PRAC\_3"  android:roundIcon="@mipmap/ic\_launcher\_round"  android:supportsRtl="true"  android:theme="@style/AppTheme">  <activity android:name=".MainActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />  <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application>  </manifest> |

**OUTPUT:**



Main View Of app



**Description**: The output of this application is shown in the image

It will Toast the message at the time of 10 second

If you have to change then change in src/MainActivity.java in that change the value of I from -1 to your value for changing message change in “Toast” method.**LATEST APPLICATIONS:**

Android Studio, ADB (Android Debug Bridge), AVD Manager, Eclipse, Fabric, Genymotion,

**LEARNING OUTCOME:**

1.We learn how to make Toast message in android application and also

2.We learned how to use Chronometer class to add timer and at the specific time interval using OnChronometerTickListener() function we can generate the toast meassage