PRACTICAL – 9

Create an application that record the audio file and storing it in the external directory in 3gp format. Also, add a button which will be able to play the same recorded sound.

<https://github.com/18mca8021/>

Activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/ic\_launcher\_foreground"**>  
  
 <**Button  
 android:id="@+id/btnRecord"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="100dp"  
 android:layout\_marginTop="120dp"  
 android:foreground="@drawable/start"** />  
  
 <**Button  
 android:id="@+id/btnStop"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="100dp"  
 android:foreground="@drawable/stop"** />  
  
 <**Button  
 android:id="@+id/btnPlay"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="100dp"  
 android:foreground="@drawable/playpause"** />  
  
 <**Button  
 android:id="@+id/btnStopPlay"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="100dp"  
 android:foreground="@drawable/pause"** />  
  
 <**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="SOUND RECORDER"  
 android:textSize="30dp"  
 android:layout\_marginTop="-260dp"  
 android:layout\_marginHorizontal="50dp"**/>  
  
  
</**LinearLayout**>

MainActivity.java

**package** com.example.myapplication9;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.core.app.ActivityCompat;  
**import** androidx.core.content.ContextCompat;  
  
**import** android.content.pm.PackageManager;  
**import** android.media.MediaPlayer;  
**import** android.media.MediaRecorder;  
**import** android.os.Bundle;  
**import** android.os.Environment;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.Toast;  
  
**import** java.io.IOException;  
  
**import static** android.Manifest.permission.***RECORD\_AUDIO***;  
**import static** android.Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** Button **startbtn**, **stopbtn**, **playbtn**, **stopplay**;  
 **private** MediaRecorder **mRecorder**;  
 **private** MediaPlayer **mPlayer**;  
 **private static final** String ***LOG\_TAG*** = **"AudioRecording"**;  
 **private static** String *mFileName* = **null**;  
 **public static final int *REQUEST\_AUDIO\_PERMISSION\_CODE*** = 1;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate( savedInstanceState );  
 setContentView( R.layout.***activity\_main*** );  
 **startbtn** = (Button)findViewById(R.id.***btnRecord***);  
 **stopbtn** = (Button)findViewById(R.id.***btnStop***);  
 **playbtn** = (Button)findViewById(R.id.***btnPlay***);  
 **stopplay** = (Button)findViewById(R.id.***btnStopPlay***);  
 **stopbtn**.setEnabled(**false**);  
 **playbtn**.setEnabled(**false**);  
 **stopplay**.setEnabled(**false**);  
 *mFileName* = Environment.*getExternalStorageDirectory*().getAbsolutePath();  
 *mFileName* += **"/AudioRecording.3gp"**;  
  
 **startbtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if**(CheckPermissions()) {  
 **stopbtn**.setEnabled(**true**);  
 **startbtn**.setEnabled(**false**);  
 **playbtn**.setEnabled(**false**);  
 **stopplay**.setEnabled(**false**);  
 **mRecorder** = **new** MediaRecorder();  
 **mRecorder**.setAudioSource(MediaRecorder.AudioSource.***MIC***);  
 **mRecorder**.setOutputFormat(MediaRecorder.OutputFormat.***THREE\_GPP***);  
 **mRecorder**.setAudioEncoder(MediaRecorder.AudioEncoder.***AMR\_NB***);  
 **mRecorder**.setOutputFile(*mFileName*);  
 **try** {  
 **mRecorder**.prepare();  
 } **catch** (IOException e) {  
 Log.*e*(***LOG\_TAG***, **"prepare() failed"**);  
 }  
 **mRecorder**.start();  
 Toast.*makeText*(getApplicationContext(), **"Recording Started"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else** {  
 RequestPermissions();  
 }  
 }  
 });  
 **stopbtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **stopbtn**.setEnabled(**false**);  
 **startbtn**.setEnabled(**true**);  
 **playbtn**.setEnabled(**true**);  
 **stopplay**.setEnabled(**true**);  
 **mRecorder**.stop();  
 **mRecorder**.release();  
 **mRecorder** = **null**;  
 Toast.*makeText*(getApplicationContext(), **"Recording Stopped"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 **playbtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **stopbtn**.setEnabled(**false**);  
 **startbtn**.setEnabled(**true**);  
 **playbtn**.setEnabled(**false**);  
 **stopplay**.setEnabled(**true**);  
 **mPlayer** = **new** MediaPlayer();  
 **try** {  
 **mPlayer**.setDataSource(*mFileName*);  
 **mPlayer**.prepare();  
 **mPlayer**.start();  
 Toast.*makeText*(getApplicationContext(), **"Recording Started Playing"**, Toast.***LENGTH\_SHORT***).show();  
 } **catch** (IOException e) {  
 Log.*e*(***LOG\_TAG***, **"prepare() failed"**);  
 }  
 }  
 });  
 **stopplay**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **mPlayer**.release();  
 **mPlayer** = **null**;  
 **stopbtn**.setEnabled(**false**);  
 **startbtn**.setEnabled(**true**);  
 **playbtn**.setEnabled(**true**);  
 **stopplay**.setEnabled(**false**);  
 Toast.*makeText*(getApplicationContext(),**"Playing Audio Stopped"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 }  
 @Override  
 **public void** onRequestPermissionsResult(**int** requestCode, String[] permissions, **int**[] grantResults) {  
 **switch** (requestCode) {  
 **case *REQUEST\_AUDIO\_PERMISSION\_CODE***:  
 **if** (grantResults.**length**> 0) {  
 **boolean** permissionToRecord = grantResults[0] == PackageManager.***PERMISSION\_GRANTED***;  
 **boolean** permissionToStore = grantResults[1] == PackageManager.***PERMISSION\_GRANTED***;  
 **if** (permissionToRecord && permissionToStore) {  
 Toast.*makeText*(getApplicationContext(), **"Permission Granted"**, Toast.***LENGTH\_LONG***).show();  
 } **else** {  
 Toast.*makeText*(getApplicationContext(),**"Permission Denied"**,Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 **break**;  
 }  
 }  
 **public boolean** CheckPermissions() {  
 **int** result = ContextCompat.*checkSelfPermission*(getApplicationContext(), ***WRITE\_EXTERNAL\_STORAGE***);  
 **int** result1 = ContextCompat.*checkSelfPermission*(getApplicationContext(), ***RECORD\_AUDIO***);  
 **return** result == PackageManager.***PERMISSION\_GRANTED*** && result1 == PackageManager.***PERMISSION\_GRANTED***;  
 }  
 **private void** RequestPermissions() {  
 ActivityCompat.*requestPermissions*(MainActivity.**this**, **new** String[]{***RECORD\_AUDIO***, ***WRITE\_EXTERNAL\_STORAGE***}, ***REQUEST\_AUDIO\_PERMISSION\_CODE***);  
 }  
}

Output







