**MainActivity**

package com.example.charan.audiorecord;

import android.media.MediaPlayer;

import android.media.MediaRecorder;

import android.os.Environment;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import java.io.IOException;

import java.util.Random;

import static android.Manifest.permission.RECORD\_AUDIO;

import static android.Manifest.permission.WRITE\_EXTERNAL\_STORAGE;

import android.support.v4.app.ActivityCompat;

import android.content.pm.PackageManager;

import android.support.v4.content.ContextCompat;

public class MainActivity extends AppCompatActivity {

Button buttonStart, buttonStop, buttonPlayLastRecordAudio,

buttonStopPlayingRecording ;

String AudioSavePathInDevice = null;

MediaRecorder mediaRecorder ;

Random random ;

String RandomAudioFileName = "ABCDEFGHIJKLMNOP";

public static final int RequestPermissionCode = 1;

MediaPlayer mediaPlayer ;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

buttonStart = (Button) findViewById(R.id.button);

buttonStop = (Button) findViewById(R.id.button2);

buttonPlayLastRecordAudio = (Button) findViewById(R.id.button3);

buttonStopPlayingRecording = (Button)findViewById(R.id.button4);

buttonStop.setEnabled(false);

buttonPlayLastRecordAudio.setEnabled(false);

buttonStopPlayingRecording.setEnabled(false);

random = new Random()

buttonStart.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if(checkPermission()) {

AudioSavePathInDevice =

Environment.getExternalStorageDirectory().getAbsolutePath() + "/" +

CreateRandomAudioFileName(5) + "AudioRecording.3gp";

MediaRecorderReady();

try {

mediaRecorder.prepare();

mediaRecorder.start();

} catch (IllegalStateException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

buttonStart.setEnabled(false);

buttonStop.setEnabled(true);

Toast.makeText(MainActivity.this, "Recording started",

Toast.LENGTH\_LONG).show();

} else {

requestPermission();

}

}

});

buttonStop.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

mediaRecorder.stop();

buttonStop.setEnabled(false);

buttonPlayLastRecordAudio.setEnabled(true);

buttonStart.setEnabled(true);

buttonStopPlayingRecording.setEnabled(false);

Toast.makeText(MainActivity.this, "Recording Completed",

Toast.LENGTH\_LONG).show();

}

});

buttonPlayLastRecordAudio.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) throws IllegalArgumentException,

SecurityException, IllegalStateException {

buttonStop.setEnabled(false);

buttonStart.setEnabled(false);

buttonStopPlayingRecording.setEnabled(true);

mediaPlayer = new MediaPlayer();

try {

mediaPlayer.setDataSource(AudioSavePathInDevice);

mediaPlayer.prepare();

} catch (IOException e) {

e.printStackTrace();

}

mediaPlayer.start();

Toast.makeText(MainActivity.this, "Recording Playing",

Toast.LENGTH\_LONG).show();

}

});

buttonStopPlayingRecording.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

buttonStop.setEnabled(false);

buttonStart.setEnabled(true);

buttonStopPlayingRecording.setEnabled(false);

buttonPlayLastRecordAudio.setEnabled(true);

if(mediaPlayer != null){

mediaPlayer.stop();

mediaPlayer.release();

MediaRecorderReady();

}

}

});

}

public void MediaRecorderReady(){

mediaRecorder=new MediaRecorder();

mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);

mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE\_GPP);

mediaRecorder.setAudioEncoder(MediaRecorder.OutputFormat.AMR\_NB);

mediaRecorder.setOutputFile(AudioSavePathInDevice);

}

public String CreateRandomAudioFileName(int string){

StringBuilder stringBuilder = new StringBuilder( string );

int i = 0 ;

while(i < string ) {

stringBuilder.append(RandomAudioFileName.

charAt(random.nextInt(RandomAudioFileName.length())));

i++ ;

}

return stringBuilder.toString();

}

private void requestPermission() {

ActivityCompat.requestPermissions(MainActivity.this, new

String[]{WRITE\_EXTERNAL\_STORAGE, RECORD\_AUDIO}, RequestPermissionCode);

}

@Override

public void onRequestPermissionsResult(int requestCode,

String permissions[], int[] grantResults) {

switch (requestCode) {

case RequestPermissionCode:

if (grantResults.length> 0) {

boolean StoragePermission = grantResults[0] ==

PackageManager.PERMISSION\_GRANTED;

boolean RecordPermission = grantResults[1] ==

PackageManager.PERMISSION\_GRANTED;

if (StoragePermission && RecordPermission) {

Toast.makeText(MainActivity.this, "Permission Granted",

Toast.LENGTH\_LONG).show();

} else {

Toast.makeText(MainActivity.this,"PermissionDenied",Toast.LENGTH\_LONG).show();

}

}

break;

}

}

public boolean checkPermission() {

int result = ContextCompat.checkSelfPermission(getApplicationContext(),

WRITE\_EXTERNAL\_STORAGE);

int result1 = ContextCompat.checkSelfPermission(getApplicationContext(),

RECORD\_AUDIO);

return result == PackageManager.PERMISSION\_GRANTED &&

result1 == PackageManager.PERMISSION\_GRANTED;

}

}

**XML**

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Record"

android:id="@+id/button"

android:layout\_below="@+id/imageView"

android:layout\_alignParentLeft="true"

android:layout\_marginTop="37dp"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="STOP"

android:id="@+id/button2"

android:layout\_alignTop="@+id/button"

android:layout\_centerHorizontal="true"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Play"

android:id="@+id/button3"

android:layout\_alignTop="@+id/button2"

android:layout\_alignParentRight="true"

android:layout\_alignParentEnd="true"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="STOP PLAYING RECORDING "

android:id="@+id/button4"

android:layout\_below="@+id/button2"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="10dp"

/>

</RelativeLayout>

**Manifest**

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

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

package="com.example.charan.audiorecord" >

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

<uses-permission android:name="android.permission.RECORD\_AUDIO" />

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

<uses-sdk

android:minSdkVersion="8"

android:targetSdkVersion="17" />

<application

android:allowBackup="true"

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

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

android:theme="@style/AppTheme" >

<activity

android:name=".MainActivity"

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

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

</manifest>