If you don’t want any error create file and folder name same as I do

**Step 1 Setup project with folder name MediaPlayerApp**

**Step 2.** look for res folder -> right click on res -> new -> sample data factory -> set value raw ->

now copy mp3 music and paste it on raw directories file name (music.mp3)

**Step 3.** Inside manifests file -> AndroidManifest.xml add following code

**===================================================================**

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

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

package="com.example.mediaplayerapp">

<!-- Add permission for foreground service media playback -->

<uses-permission android:name="android.permission.FOREGROUND\_SERVICE\_MEDIA\_PLAYBACK" />

<uses-permission android:name="android.permission.FOREGROUND\_SERVICE" />

<application

android:allowBackup="true"

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

android:theme="@style/Theme.MediaPlayerApp">

<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>

<!-- Add foregroundServiceType attribute -->

<service

android:name="com.example.mediaplayerapp.MusicService"

android:foregroundServiceType="mediaPlayback" /> <!-- Add this line -->

</application>

</manifest>

**4. Inside activity\_main.xml add following code**

===================================================================================

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

<LinearLayout

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:gravity="center"

android:padding="16dp">

<Button

android:id="@+id/startBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Start Music" />

<Button

android:id="@+id/stopBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Stop Music"

android:layout\_marginTop="16dp" />

</LinearLayout>

**Step 5. Inside MainActivity add following code**

================================================================

package com.example.mediaplayerapp

import android.content.Intent

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import android.widget.Button

import com.example.mediaplayerapp.MusicService

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val startBtn = findViewById<Button>(R.id.startBtn)

val stopBtn = findViewById<Button>(R.id.stopBtn)

val serviceIntent = Intent(this, MusicService::class.java)

startBtn.setOnClickListener {

startService(serviceIntent)

}

stopBtn.setOnClickListener {

stopService(serviceIntent)

}

}

}

Step 6. Create MUSICSERVICE.kt file inside com.example.mediaplayer

============ Add Following Code ====================

package com.example.mediaplayerapp

import android.app.Notification

import android.app.NotificationChannel

import android.app.NotificationManager

import android.app.Service

import android.content.Intent

import android.media.MediaPlayer

import android.os.Build

import android.os.IBinder

import android.util.Log

import android.widget.Toast

import androidx.core.app.NotificationCompat

import com.example.mediaplayerapp.R

class MusicService : Service() {

private var mediaPlayer: MediaPlayer? = null

private val CHANNEL\_ID = "MusicServiceChannel"

override fun onCreate() {

super.onCreate()

Log.d("MusicService", "Service created")

}

override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {

Log.d("MusicService", "Service started")

Toast.makeText(this, "Service Started", Toast.LENGTH\_SHORT).show()

// Create a notification channel for foreground service (for Android O and above)

createNotificationChannel()

// Create a notification for foreground service

val notification: Notification = NotificationCompat.Builder(this, CHANNEL\_ID)

.setContentTitle("Music Playing")

.setContentText("Your background music is playing")

.setSmallIcon(android.R.drawable.ic\_media\_play)

.build()

// Start the service as a foreground service

startForeground(1, notification)

// Initialize and start MediaPlayer only if it's not already started

if (mediaPlayer == null) {

try {

mediaPlayer = MediaPlayer.create(this, R.raw.music) // Ensure you have "simple.mp3" in res/raw

mediaPlayer?.isLooping = true

mediaPlayer?.start()

Log.d("MusicService", "MediaPlayer started")

} catch (e: Exception) {

Log.e("MusicService", "MediaPlayer failed: ${e.message}")

}

}

return START\_STICKY

}

override fun onDestroy() {

super.onDestroy()

// Properly release MediaPlayer when the service is stopped

mediaPlayer?.apply {

stop()

release()

}

mediaPlayer = null

Toast.makeText(this, "Service Stopped", Toast.LENGTH\_SHORT).show()

Log.d("MusicService", "Service destroyed")

}

override fun onBind(intent: Intent?): IBinder? = null

// Create a notification channel for Android O and above

private fun createNotificationChannel() {

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

val serviceChannel = NotificationChannel(

CHANNEL\_ID,

"Music Service Channel",

NotificationManager.IMPORTANCE\_LOW

)

val manager = getSystemService(NotificationManager::class.java)

manager.createNotificationChannel(serviceChannel)

}

}

}