|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Advanced Mobile Programming  Practical #10 | | |
|  |  |  |  |
| **Name** | Aniket Gupta | **Roll Number** | 21302B0053 |
| **Subject/Course:** | Advanced Mobile Programming | | |
| **Topic** | Programming Media API and Telephone API | | |
|  | | | |
| **Media API** | | | |
| 10. a) To create a program to access music (media) in your mobile phone. | | | |
| **MainActivity.kt**  package com.example.amp\_p10\_11  import android.media.MediaPlayer import android.os.Bundle import android.widget.Button import androidx.activity.ComponentActivity import android.os.Handler  class MainActivity : ComponentActivity() {  private lateinit var mediaPlayer: MediaPlayer  private lateinit var playButton: Button  private lateinit var pauseButton: Button  private lateinit var stopButton: Button   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)   playButton = findViewById(R.id.*playButton*)  pauseButton = findViewById(R.id.*pauseButton*)  stopButton = findViewById(R.id.*stopButton*)   mediaPlayer = MediaPlayer.create(this, R.raw.*music*)    playButton.setOnClickListener **{** mediaPlayer.start()  updateSeekBar()  **}**  pauseButton.setOnClickListener **{** if (mediaPlayer.*isPlaying*) {  mediaPlayer.pause()  }  **}**  stopButton.setOnClickListener **{** mediaPlayer.stop()  mediaPlayer.release()  **}** }   private fun updateSeekBar() {  val handler = Handler()  handler.postDelayed(object : Runnable {  override fun run() {  if (mediaPlayer.*isPlaying*) {   handler.postDelayed(this, 1000)  }  }  }, 1000)  }   override fun onDestroy() {  super.onDestroy()  mediaPlayer.release()  } }  **AndroidManifest.xml**  <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/> <uses-permission android:name="android.permission.WAKE\_LOCK"/> <uses-permission android:name="android.permission.INTERNET" />  **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:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">    <Button  android:id="@+id/playButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_centerInParent="true"  android:backgroundTint="#4BD651"  android:text="Play" />   <Button  android:id="@+id/pauseButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_below="@id/playButton"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="16dp"  android:backgroundTint="#1984DA"  android:text="Pause" />   <Button  android:id="@+id/stopButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_below="@id/pauseButton"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="16dp"  android:backgroundTint="#C11313"  android:text="Stop" />  </RelativeLayout>  **OUTPUT:** | | | |
|  | | | |
| **Telephone API** | | | |
| 10.b) To create a program that uses the calling feature of android mobile phones. | | | |
| **MainActivity.kt**  package com.example.amp\_7\_1  import android.content.Intent import android.content.pm.PackageManager import android.net.Uri import android.support.v7.app.AppCompatActivity import android.os.Bundle import android.support.v4.app.ActivityCompat import android.widget.Button  class MainActivity : AppCompatActivity() {  val mobno:String="8888888888"  val REQUEST\_PHONE\_CALL=1  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.activity\_main)  val call=findViewById<Button>(R.id.phonecall)  call.setOnClickListener**{** if(ActivityCompat.checkSelfPermission(this,android.Manifest.permission.*CALL\_PHONE*)!=PackageManager.*PERMISSION\_GRANTED*)  {  ActivityCompat.requestPermissions(this, *arrayOf*(android.Manifest.permission.*CALL\_PHONE*),REQUEST\_PHONE\_CALL)  }  else  {  makecall()  }  **}** }  private fun makecall(){  val intent=Intent(Intent.*ACTION\_CALL*, Uri.fromParts("tel",mobno,null))  startActivity(intent)  }   override fun onRequestPermissionsResult(  requestCode: Int,  permissions: Array<out String>,  grantResults: IntArray  ) {  if (requestCode==REQUEST\_PHONE\_CALL)  {  makecall()  }  }  }  **activity\_main.xml**  <?xml version="1.0" encoding="utf-8"?> <android.support.constraint.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"  tools:context=".MainActivity">   <Button  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Place Call"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  android:id="@+id/phonecall"  android:layout\_marginTop="200dp"  app:layout\_constraintTop\_toTopOf="parent" />  </android.support.constraint.ConstraintLayout>  **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"  package="com.example.amp\_6\_1">  <uses-permission android:name="android.permission.CALL\_PHONE"></uses-permission>  <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/Theme.AMP\_6\_1](mailto:android:theme="@style/Theme.AMP_6_1)"  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>  **OUTPUT:** | | | |